# Cost-Efficient Multicast Routing in Ad Hoc and Sensor Networks

Pedro M. Ruiz,* DIIC, University of Murcia, pedrom@dif.um.es

Ivan Stojmenovic, SITE, University of Ottawa, ivan@site.uottawa.ca

August 18, 2005

## 1 Introduction

A mobile ad hoc network (MANET) consists of a number of devices equipped with wireless interfaces. Ad hoc nodes are free to move, and communicate with each other using their wireless interfaces. Communications among nodes which are not within the same radio range are carried on via multihop routing. That is, some of the intermediate nodes between the source and the destination act as relays to deliver the messages. Hence, these networks can be deployed without any infrastructure, making them specially interesting for dynamic scenarios like battlefield, rescue operations and even as flexible extensions of mobile networks for operators.

Wireless sensor networks (WSNs) follow a similar communication paradigm based on multihop paths. Although wireless sensor nodes are not usually mobile, their limited resources in terms of battery life and computational power pose additional challenges to the routing task. For instance, wireless sensor nodes operate following a duty-cycle, allowing them to save energy while they are sleeping. The different timings for sleep and awake periods across sensors makes the topology change. In addition, these networks are usually densely populated compared to ad hoc networks, requiring very efficient and scalable mechanisms to provide the routing functions. Examples of such techniques gaining momentum nowadays are geographic routing and localized algorithms in general, in which nodes take individual decisions solely based on the local information about itself and its neighbors. These networks have a lot of potential applications, which is one of the reasons why they are receiving so much attention within the research community.

*Work supported by the "Ramon y Cajal" workprogramme from the Spanish Education Ministry

The most commonly used model for these networks is called "Unit Disk Graph" . The network is modeled as an undirected graph $G = (V, E)$ where $V$ is the set of vertices and $E$ is the set of edges. The model assumes that the network is two dimensional (every node $v \in V$ is embedded in the plane) and wireless nodes are represented by vertices of the graph. Each node $v \in V$ has a transmission range $r$. Let $dist(v_1, v_2)$ be the distance between two vertices $v_1, v_2 \in V$. An edge between two nodes $v_1, v_2 \in V$ exists iff $dist(v_1, v_2) \leq r$ (i.e. $v_1$ and $v_2$ are able to communicate directly).

Unicast routing both for MANETs and WSNs can be defined as the process of finding a paths in the network to deliver a message from the originator to the destination. As we mentioned before, in these networks such paths are formed by a set of nodes acting as relays. The multicast routing task is similar the unicast routing except that there are a number of destinations instead of a single node. These destinations are often referred as "receivers" in the literature. In this particular case, the set of relay nodes usually forms a tree, commonly known as "multicast tree". Below, we define more precisely the problem of unicast and multicast routing in these networks.

**Definition 56.1** *Given a graph $G = (V, E)$, a source node $s \in V$ and a destination node $D \in V$, the unicast routing problem can be defined as finding a set of relay nodes $F \subset V$ s.t. $\{s\} \cup F \cup \{D\}$ is connected.*

Similarly, the multicast routing problem can be defined as follows:

**Definition 56.2** *Given a graph $G = (V, E)$, a source node $s \in V$ and a set of destinations $R \subseteq V$, the multicast routing problem, can be defined as finding a set of relay nodes $F \subset V$ s.t. $\{s\} \cup F \cup R$ is connected.*

Of course, routing algorithms are designed to avoid cycles, and usually select paths according to some metric or combination of metrics such as hop count, delays, etc. In fact, most of the existing routing protocols use the hop count as the path selection metric.

The problem of unicast routing is well-known and there are many distributed algorithms like Dijkstra, Bellman-Ford, etc. For the problem of multicast routing, there are also algorithms to build source path trees (SPT), shared trees, etc. In fact, the problem of the efficient distribution of traffic from a set of senders to a group of receivers in a datagram network was already studied by Deering [1] in the late 80's. Several multicast routing protocols like DVMRP [2], MOSPF [3], CBT [4] and PIM [5] have been proposed for IP multicast routing in fixed networks. These protocols have not been usually considered in mobile ad hoc networks because they do not properly support mobility. In the case of mesh networks, one may believe that they can be a proper solution. However, they were not designed to operate on wireless links, and they lead to sub-optimal routing solutions which are not able to take advantage of the broadcast nature of the wireless medium (i.e. sending a single message to forward a multicast message to all the next hops rather than replicating the message for each neighbor). Moreover, their routing metrics do not aim at minimizing the cost of the multicast tree, which limits the overall capacity of the mesh network.

Within the next sections, we describe existing multicast routing protocols for ad hoc and sensor networks, and we analyze the issue of computing minimum cost multicast trees. In fact, we will show the NP-completeness of the problem.

Given that the use of approximation algorithms is fully justified in the multicast routing case, we focus the rest of the chapter on the multicast routing problem, and its approximation algorithms for MANETs and WSNs. The remainder of the chapter is organized as follows: Section 2 describes existing multicast routing protocols for MANETs, and their inability to approximate minimum cost multicast trees. Section 3 focus on the issue of computing minimum bandwidth multicast trees in wireless ad hoc networks, shows the NP-completeness of the problem, and offers approximation algorithms which offer better performance than Steiner trees. We focus on the problem of geographic multicast routing in Section 4. Finally, we provide some discussion and conclusions in Section 5.

# 2 Multicast routing in ad hoc networks

A plethora of protocols have been proposed for multicast routing in mobile ad hoc networks. We focus our discussion on the most representative protocols. They can be classified into tree or mesh-based depending upon the underlying forwarding structure that they use. Tree-based schemes [6, 7, 8, 9, 12] construct a multicast tree from each of the sources to all the receivers using generally a source path tree, or a shared tree. Mesh-based approaches [10, 11], compute several paths among senders and destinations. Thus, when the mobility rate increases they are able to tolerate link breaks better than tree-based protocols at the expense of a usually higher overhead. Hybrid approaches [13, 14] try to combine the robustness of mesh-based ad hoc routing and the low overhead of tree-based protocols. Finally, there are stateless multicast protocols [15] in which there is no need to maintain a forwarding state on the nodes (for instance, if the nodes to traverse are included in the data packets themselves). We will not discuss further about the latter category given the very limited applicability of those variants.

Regarding tree-based protocols, AMRIS [6] builds a shared multicast tree among a set of sources and receivers. There is a root node, which is the one with the smallest ID (Sid). These ID numbers are assigned dynamically within a multicast session, and based on these IDs the multicast tree is built. The numbering process starts at the Sid, and other nodes always select an ID being higher that the one of his upstream node in the tree. MAODV [7] is an extension of the well-known AODV protocol. The route creation is similar to the RREQ/RREP process in AODV, except that the source unicasts a MACT (Multicast Activation) message through the selected paths, which usually form a shortest path tree (SPT) based on the hop count.

Regarding mesh-based multicast routing protocols for ad hoc networks, ODMRP [10] works reactively to build a multicast mesh connecting senders and receivers. All the intermediate nodes taking part in the multicast mesh are said to belong to the forwarding group (FG). When a multicast node has data to send and it does not have a route for that multicast group, it starts a periodic broadcasting of JOIN_QUERY (JQ) packets. These messages are propagated through the entire ad hoc network avoiding duplicates, so that every ad hoc node can learn which of his neighbors is in its shortest path to that source. Upon reception of a non-duplicate JQ message, a receiver broadcasts a JOIN_REPLY (JR) message in which it includes the ID of the neighbor selected as the next hop to reach each of the multicast sources. When a node receives a JR message it checks out if its own ID is listed as the selected next hop for any of the sources. If that is the case, then it realizes that it is in the shortest path tree to any of the sources, and it adds itself to the FG by activating its FG_FLAG. In addition, the selected node sends out a JR which he fills with the IDs of its selected neighbors to reach those sources for which it was selected in the received JR message. In this way, the FG is populated until the JR messages reach the source. This whole process is repeated periodically to update the FG over time. Once the mesh is built, data forwarding is very simple. Only those nodes whose FG_FLAG is active are allowed to forward data packets generated by the sources. In addition, in case of receiving the same data packet several times, a node within the FG shall only forward it the first time it is received. CAMP [11] was designed as an extension of the "Core Based Trees" (CBT [4]) protocol. However, unlike CBT in which there was not link redundancy, CAMP builds a multicast mesh to offer a much better performance and resilience in case of link breaks. Whereas in CBT core nodes were used for data forwarding, they are used in CAMP to reduce the overhead for a node to find out a node belonging to the multicast mesh. Data packets are not required to go through core nodes. Thus, given that there can be several core nodes for a multicast group, the tolerance of mobility is increased. CAMP uses a receiver-initiated approach to build the multicast mesh. Using the cores as well-known mesh nodes, CAMP avoids relying on periodic flooding of the network to find multicast routes. However, this comes at the cost of depending upon the unicast routing protocol as well as the need of a mapping service from multicast groups to core nodes. CAMP ensures that the mesh contains all the reverse shortest paths between the source and the recipients. It uses a so-called "heartbeat" mechanism by which each node periodically monitors its packet cache. If the node finds out that some data packets which it is receiving are not coming from its shortest path neighbor, then it sends a HEARTBEAT message to its successor in the reverse shortest path to the source. The HEARTBEAT triggers a push-join (PJ) message which (if the successor is not a mesh member) forces the successor and all the nodes in the path to join the mesh.

All these solutions are not aimed at minimizing the cost of multicast trees due to the difficulty of computing such trees. In fact, when the goal is to find multicast trees with minimum edge cost, the problem becomes NP-complete and requires heuristic solutions. Such minimum cost multicast tree is well-known as the Steiner tree problem. However, Ruiz et. al [23] showed that ODMRP (and accordingly other similar protocols) can benefit from the use of Steiner trees rather than SPTs and shared trees. Karp [16] demonstrated that the Steiner tree problem is NP-complete even when every link has the same cost using a transformation from the exact cover by 3-sets. There are some heuristic algorithms [17] to compute minimal Steiner trees. For instance, the MST heuristic ([18, 19]) provides a 2-approximation, and Zelikovsky [20] proposed an algorithm which obtains a 11/6-approximation. Recently, Rajagopalan and Vazirani [21] proposed a 3/2-approximation algorithm. Given the complexity of computing this kind of trees in a distributed way, most of the existing multicast routing protocols use shortest path trees or sub-optimal shared trees, which can be easily computed in polynomial time.

Recently, Ruiz et al. [24] showed that the Steiner tree is not the best solution for wireless ad hoc networks, and provided some approximation algorithms which will be analyzed in this chapter. The problem of minimizing the bandwidth consumption of a multicast tree in an ad hoc network needs to be re-formulated in terms of minimizing the number of data transmissions. By assigning a cost to each link of the graph computing the tree which minimizes the sum of the cost of its edges, existing formulations have implicitly assumed that a given node $v$, needs $k$ transmissions to send a multicast data packet to $k$ of its neighbors. However, in a broadcast medium, the transmission of a multicast data packet from a given node $v$ to any number of its neighbors can be done with a single data transmission. Thus, in ad hoc networks the minimum cost tree is the one which connects sources and receivers by issuing a minimum number of transmissions, rather than having a minimal edge cost. We will discuss further this problem along next sections.

# 3 Minimum Bandwidth Consumption Multicast Tree

When nodes are mobile, such as in traditional MANETs, it is not really interesting to approximate optimal multicast trees. The reason is that by the time the tree is computed, it may no longer exist. However, in some new static ad hoc network scenarios being deployed (i.e. wireless mesh networks) these algorithms may become of utmost relevance. In wireless mesh networks, devices are powered. So, the main concern, rather than being the power consumption, is the proper utilization of the bandwidth. Thus, the computation of bandwidth-optimal multicast trees becomes really important.

Given a multicast source $s$ and a set of receivers $R$ in a network represented by a undirected graph, we are interested in finding the multicast tree with the minimal cost in terms of the total amount of bandwidth required to deliver a packet from $s$ to every receiver.

In wired networks, the computation of such minimum bandwidth consumption multicast tree is equivalent to the Steiner tree over a graph $G = (V, E)$ so that $w(e_i) = b_s, \forall i = 1..|E|$, being $b_s$ the rate at which the source $s$ is transmitting, and $w(e_i)$ the cost of the link number $i$. Unitary costs (i.e. $w(e_i) = 1, \forall i = 1..|E|$) are usually assumed for simplicity. The bandwidth consumption of such a Steiner tree $T^* = (V^*, E^*)$ is then proportional to $|E^*|$. This means that the problem is NP-complete in wired networks. As we mentioned before, in wireless multihop networks a node can send a message to all neighbors with a single transmission. Thus, the bandwidth consumption is different from the edge cost, and the problem requires being reformulated in terms of the number of transmissions rather than the number of traversed links. To account for the excessive bandwidth consumption due to suboptimal trees, we will define a new metric called "data overhead".

## 3.1 Problem Formulation

Before going into details, we need some definitions which are used in the sequel.

**Definition 56.3** *Given a graph $G = (V, E)$, a source $s \in V$ and a set of receivers $R \subset V$, we define the set $T$ as the set of the possible multicast trees in $G$ which connect the source $s$ to every receiver $r_i \in R$. We denote by $F_t$, the set of relay nodes in the tree $t \in T$, consisting of every non-leaf node, which relays the message sent out by the multicast source. We can define a function $C_t : T \to \mathbb{Z}^+$ so that given a tree $t \in T$, $C_t(t)$ is the number of transmissions required to deliver a message from the source to every receiver induced by that tree.*

**Lemma 56.1** *Given a tree $t \in T$ as defined above, then $C_t(t) = 1 + |F_t|$.*

**Proof**

By definition relay nodes forward the message sent out by $s$ only once. In addition, leaf nodes do not forward the message. Thus, the total number of transmissions is one from the source, and one from each relay node. Making a total of $1 + |F_t|$. □

So, as we can see from 56.1, to minimize $C_t(t)$ we must somehow reduce the number of forwarding nodes $|F_t|$. Note that some receivers may serve also as relay nodes.

**Definition 56.4** *Under the conditions of definition 56.3, let $t^* \in T$ be the multicast tree such that $C_t(t^*) \leq C_t(t)$ for any possible $t \in T$. We define the data overhead of a tree $t \in T$, as $\omega_d(t) = C_t(t) - C_t(t^*)$. Obviously, with this definition $\omega_d(t^*) = 0$.*

Based on the previous definitions, the problem can be formulated as follows. Given a graph $G = (V, E)$, a source node $s \in V$, a set of receivers $R \subset V$, and given $V' \subseteq V$ defined as $V' = R \cup \{s\}$, find a tree $T^* = (V^*, E^*) \subset G$ such that the following conditions are satisfied:

(1) $V^* \supseteq V'$

(2) $C_t(T^*)$ is minimized.

From the condition of $T^*$ being a tree it is obvious that it is connected, which combined with condition (1) establishes that $T^*$ is a multicast tree. Condition (2) is equivalent to $\omega_d(T^*) = 0$, and establishes the optimality of the tree. As we show below, this problem is NP-complete.

## 3.2 NP-completeness

Ruiz et al. [24] demonstrated that this problem is NP-complete. We show below a similar demonstration based on the inclusion of the Minimim Common Dominating Set (MCDS) as a particular case of the problem.

**Theorem 56.1** *Given a graph $G = (V, E)$, a multicast source $s \in V$ and a set of receivers $R$, the problem of finding a tree $T^* \supseteq R \cup \{s\}$ so that $C_t(T^*)$ is minimum is NP-complete.*

**Proof**

According to lemma 56.1, minimizing $C_t(T^*)$ is equivalent to minimize the number of relay nodes $F \subseteq T^*$. So, the problem is finding the smallest set of forwarding nodes $F$ that connects $s$ to every $r \in R$. If we consider the particular case in which $R = V - \{s\}$, the goal is finding the smallest connected $F \subseteq T^*$ which covers the rest of nodes in the graph $(V - \{s\})$. This problem is one of finding a minimum connected dominating set, which is known to be NP-complete [22]. □

Ruiz et al. [24] also proved the suboptimality of Steiner trees for this particular problem in ad hoc networks. We give a simple example to prove it.

**Theorem 56.2** *Let $G = (V, E)$ be an undirected graph. Let $s \in V$ be a multicast source and $R \subseteq V$ be the set of receivers. The Steiner multicast tree $T^* \subseteq G$ so that $C_e(T^*)$ is minimal may not be the minimal data-overhead multicast tree.*

**Proof**

It is immediate from the examples shown in Fig. 56.1 that the Steiner tree may not minimize the bandwidth consumption, leading to suboptimal solutions in MANETs and WSNs. □

In general, Steiner tree heuristics try to reduce the cost by minimizing the number of Steiner nodes $|\mathbb{S}^*|$. In the next section we will present our heuristics being able to reduce the bandwidth consumption of multicast trees, by just making receivers be leaf nodes in a cost effective way.

## 3.3 Heuristic Approximations

Given the NP-completeness of the problem, within the next subsections we describe two heuristic algorithms proposed by Ruiz et al. [24] to approximate minimal data-overhead multicast trees. As we learned from the demonstration of theorem 2, the best approach to reduce the data overhead is reducing the number of forwarding nodes, while increasing the number of leaf nodes. The two heuristics presented below try to achieve that trade-off.

### 3.3.1 Greedy-based heuristic algorithm

The first proposed algorithm is suited for centralized wireless mesh networks, in which the topology can be known by a single node, which computes the multicast tree.

Inspired on the results from theorem 56.2, this algorithm first systematically builds different cost-effective subtrees. The cost-effectiveness refers to the fact that a node $v$ is selected to be a forwarding node only if it covers two or more nodes. That is, if it has two or more multicast receivers as neighbors.

The algorithm shown in algorithm 1, starts by initializing the nodes to cover ('aux') to all the sources except those already covered by the source $s$. Initially the set of forwarding nodes ('MF') is empty. After the initialization, the algorithm repeats the process of building a cost-effective tree, starting with the node $v$ which covers the largest number of nodes in 'aux'. Then, $v$ is inserted into the set of forwarding nodes (MF) and it becomes a node to cover. In addition, the receivers covered by $v$ (Cov($v$)) are removed from the list of nodes to cover denoted by 'aux'. This process is repeated until all the nodes are covered, or it is not possible to find more cost-effective subtrees. In the latter case, the different subtrees are connected by a Steiner tree among their roots, which are in the list 'aux' (i.e. among the nodes which are not covered yet). For doing that one can use any Steiner tree heuristic. In our simulations we use the MST heuristic for simplicity.

**Algorithm 1** Greedy minimal data overhead algorithm MNT

MF $\leftarrow \oslash / * mcast - forwarders * /$

V $\leftarrow$ V - $\{s\}$

aux $\leftarrow$ R-Cov(s) + $\{s\}$ $/ * nodes - to - cover * /$

**repeat**

  node $\leftarrow argmax_{v \in V}(|\text{Cov}(v)|)$ s.t. Cov($v$)$\geq$2

  aux $\leftarrow$ aux-Cov($v$)+$\{v\}$

  V $\leftarrow$ V-$\{v\}$

  MF $\leftarrow$ MF + $\{v\}$

**until** aux = $\oslash$ or node = $null$

**if** V!=$\oslash$ **then**

  Build Steiner tree among nodes in aux

**end if**

### 3.3.2 Distributed Variant

The previous algorithm may be useful for some kind of networks. However, in general a distributed algorithm is preferred for wireless ad hoc networks. Hence, Ruiz et al. [24] proposed the distributed approach described below.

The previous protocol consists of two different parts: (i) construction of cost-efficient subtrees, and (ii) building a Steiner tree among the roots of the subtrees.

To build a Steiner tree among the roots of the subtrees, Ruiz assumed in the previous protocol the utilization of the MST heuristic. This is a centralized heuristic consisting of two different phases. Firstly, the algorithm builds the metric closure for the receivers on the whole graph, and then, a minimum spanning tree (MST) is computed on the metric closure. Finally, each edge in the MST is substituted by the shortest path (in the original graph) between the two nodes connected by that edge. Unfortunately, the metric closure of a graph is hard to build in a distributed way. Thus, Ruiz approximated the MST heuristic with the simple, yet powerful, algorithm presented in Algorithm 2. The source, or the root of the subtree in which the source is (called source-root) will start flooding a route request message (RREQ). Intermediate nodes, when propagating that message will increase the hop count. When the RREQ is received by a root of a subtree, it sends a route reply (RREP) back through the path which reported the lowest hop count. Those nodes in that path are selected as multicast forwarders (MF). In addition, a root of a subtree, when propagating the RREQ will reset the hop count field. This is what makes the process very similar to the computation of the MST on the metric closure. In fact, we achieve the same effect, which is that each root of the subtrees, will add to the Steiner tree the path from itself to the source-root, or the nearest root of a subtree. In the case in which two neighboring nodes are far away from S but at the same hop count, the node-ID is used as a tie-breaker. This avoids a deadlock by preventing each of them from calling the other as its selected next hop. The one with the lowest ID will always select the other. This mechanism and the way in which the algorithm is executed from the source-root to the other nodes guarantees that the obtained tree is connected.

The second part of the algorithm to make distributed is the creation of the cost-effective subtrees. However, this part is much simpler and it can be done locally with just a few messages. Receivers flood a Subtree_Join (ST_JOIN) message only to its 1-hop neighbors indicating the multicast group to join. These neighbors answer with a Subtree_Join_Ack (ST_ACK) indicating the number of receivers they cover. This information is known locally by counting the number of (ST_JOIN) messages received. Finally, receivers send again a Subtree_Join_Activation (ST_JOIN_ACT) message including their selected root, which is the neighbor which covers the greatest number of receivers. This is also known locally from the information in the (ST_ACK). Those nodes which are selected by any receiver, repeat the process acting as receivers. Nodes which already selected a root do not answer this time to ST_JOIN messages.

## 3.4 Performance evaluation

Ruiz et al. [24] presented some performance evaluation of their approach. We have reproduced their simulations with the same parameters except that the number of nodes has been fixed at 600, and the area has been ranged from 750x750m to 2250x2250 m. The simulated algorithms are the two minimum bandwidth algorithms (MNT and MNT2 respectively) as well as the MST heuristic to approximate Steiner trees. In addition, we also simulated the shortest path tree (SPT) algorithm, which is the one which is used by most

**Algorithm 2** Distributed approximation of MST heuristic MNT2

```
if thisnode.id = source − root then
    Send RREQ with RREQ.hopcount=0
end if
if rcvd non duplicate RREQ with better hopcount then
    prevhop ← RREQ.sender
    RREP.nexthop ← prevhop
    RREQ.sender ← thisnode.id
    if thisnode.isroot then
        send(RREP)
        RREQ.hopcount ← 0
    else
        RREQ.hopcount++;
    end if
    send(RREQ)
end if
if received RREP and RREP.nexthop = thisnode.id then
    Activate MF_FLAG
    RREP.nexthop ← prevhop
    send(RREP)
end if
```

multihop multicast routing protocols proposed to date.

Performance metrics are also the same as Ruiz et al. used in [24] and the reader can refer there for details. The results shown correspond to the 1250x1250m area and 150 receivers. For each combination of simulation parameters, a total of 91 simulation runs with different randomly-generated graphs were performed. The error in the graphs shown below are obtained using a 95% confidence level.

#### 3.4.1 Performance Analysis

In Fig. 56.2 we show for a network with an intermediate density (600 nodes in 1250x1250m area) how the number of transmissions required varies with respect to the number of receivers. For a low number of them, the minimum bandwidth schemes do not offer significant differences compared to the Steiner tree heuristic. This is because receivers tend to be very sparse and it is less likely that cost-effective trees are built. However, as the number of receivers increases, the creation of cost-effective trees is favored, making the MNT and MNT2 algorithms achieve significant reductions in the number of transmissions required. In addition, given that the SPT approach doesn't aim at minimizing the cost of the trees, it shows a lower performance compared to any of the other approaches. The distributed MNT2 algorithm, by not using the metric closure, gets a slightly lower performance compared to the centralized approach. However, both of them have very similar performance, which allow them to offer substantial bandwidth savings compared to the Steiner tree (i.e. MST heuristic).

In Fig. 56.3 we represent the mean path length. MNT and MNT2 offer a higher mean path length because grouping paths for several receivers requires deviating from their shortest paths for some of the receivers. As we can see, this metric is much more variable to the number of receivers than the number of transmissions was for the heuristic approaches. This is why the error bars are reporting a larger confidence interval for MST, MNT and MNT2.

In Fig. 56.4 we present the variation of the number of transmissions as the density varies, for 150 receivers. As the figure depicts, the higher the density, the better is the performance of all the approaches. The reason is that higher densities imply shorter path length (note that number of nodes is fixed). So in general one can reach the receivers with less number of transmissions regardless of the routing scheme. However, if we compare the performance across approaches, we can see that the reduction in the number of transmissions achieved by MNT and MNT2 is higher as the density increases. This can be easily explained by the fact that for higher densities it is more likely that several receivers can be close to the same node, which facilitates the creation of cost-effective subtrees.

We can observe that the number of receivers has small impact on the performance comparison compared to that of the density of the network.

# 4 Cost-Efficient Geographic Multicast Routing

Routing in sensor networks differ from routing in ad hoc networks. Position information is almost intrinsic to wireless sensor networks. In fact, measurements providing from sensor nodes do not usually have proper meaning unless the geographical information regarding that sensor is also reported. Position information, if available to each sensor, also simplifies routing task. This approach is commonly-known as "geographic routing". A survey of position based routing schemes is given in [25].

The general idea is very simple. Given a source $s$ and a destination $d$ (the destination is normally a fixed sink whose location is known to all sensors), the source $s$ sends the data packet to be delivered, to the neighbor which is closest to the destination. This neighbor will repeat the process again, until the message is eventually delivered to the destination. For the case in which the algorithm reaches a local minima (i.e. there is no neighbor making progress towards the destination) Bose, Morin, Stojmenovic and Urrutia proposed a recovery scheme [26] called "Face routing" which guarantees delivery.

Similarly to ad hoc network routing, different metrics can be minimized when finding the route towards a destination using geographic routing. The most used one is the reduction of the hop count. However, there are also proposals to reduce the power consumption [27], delay, etc. Stojmenovic et. al [28] proposed a general framework to optimize different metrics. The rest of the chapter will be devoted to the explanation of that framework, and its application to the problem of efficient geographic multicast routing.

## 4.1 The Cost over Progress Framework

A frequent solution when dealing with multiple metrics in geographic routing in the literature, is to introduce additional parameters, not present in the problem formulation, as part of the solution protocol. For instance, the neighbor selection function is changed to something like:

$$\alpha(metric_1) + \beta(metric_2) + \cdots + \omega(metric_n)$$

where Greek letters are the additional parameters considered, and $metric_i$ are the different metrics (e.g. delay, hop count, power consumption, etc.).

So, the performance of such a protocol often depends on the particular values for the set of parameters. In most cases, the optimal values for these parameters depend on global network conditions, which may be beyond the knowledge available to tiny sensors. In other cases, the computational and communication cost required to obtain such information is higher than the benefits provided by the particular protocol. Another typical approach is to use thresholds for the solutions, which has the effect of eliminating certain options in the protocols which may lead to suboptimal solutions, or to failures.

To avoid those issues, a simple but elegant solution is to use as the neighbor selection function, a cost over progress ratio. The idea is to optimize the ratio of operation costs (in terms of the particular metrics considered in the problem statement) and progress made (e.g. reduction in distance to the destination in the case of routing). For network coverage problems, the same concept can by applied by redefining the cost function, and considering an appropriate progress function (e.g. additional area covered). To better understand this idea, we will give a couple of examples of its application.

**Example 56.1** *We consider the case in which we want to provide geographic routing with minimal power consumption. We consider Fig. 56.5 as a reference, where $C$ is the source, $D$ the destination, and $A$ is a candidate neighbor and $A'$ is the projection of $A$ on $\overline{CD}$. In addition, we have that $|\overline{CD}| = c$, $|\overline{AD}| = a$ and $|\overline{CA}| = r$.*

*The power needed to send a message from $C$ to $A$ is proportional to $r^\alpha + k$, where $\alpha$ is the power attenuation factor ($2 \leq \alpha \leq 6$), while $k$ is a constant ($k > 0$) that accounts for running circuits at the transmitter and receiver nodes. In our framework, this power can be used as the cost measure. The progress* *can be measured according to Fig. 56.5 as* $c - a$*. Therefore, the neighbor that minimizes* $(r^\alpha + k)/(c - a)$ *is the one to be selected.*

In the next section, we discuss how the same framework can be applied to the geographic multicast routing problem.

## 4.2 Geographic multicasting with cost over progress

We now focus on the multicasting problem represented in Fig. 56.6. A source node $C$ wishes to send a packet to several destinations (sinks) with known positions $(D_1, \ldots, D_n)$. It is assumed that the number of such destinations is small, which is reasonable for the scenario in which a sensor reports to several sinks.

Mauve et al. [29] proposed a geographic multicast protocol which considers the total hop count as the metric to optimize, and distances from neighbors to destinations as part of the criterion to optimize. The impact of each of these aspects in the final neighbor selection is controlled by an external parameter $(\lambda)$ whose best value is to be separately determined. We describe below how the same problem can be solved with the cost over ratio framework, without the need for such additional parameters.

Let's assume a node $C$, after receiving a multicast message including the positions of the destinations $D_1, D_2, \ldots, D_k$. And lets also assume that $C$ evaluates neighbors $A_1, A_2, \ldots A_m$ for forwarding. If there is one neighbor which is closer to all destinations than $C$, then it may happen that there is only one next hop selected. However, it may also happen that the multicast routing task needs to be split across multiple neighbors, each handling a subset of destinations.

Consider the case in Fig 56.6 as illustration of the general principle. The current total distance to deliver the message from $C$ to all receivers is $T_1 = |\overline{CD_1}| + |\overline{CD_2}| + |\overline{CD_3}| + |\overline{CD_4}| + |\overline{CD_5}|$. If $C$ now considers neighbors $A_1$ and $A_2$ as forwarding nodes covering $D_1, D_2, D_3$ and $D_4, D_5$ respectively, the new total remaining distance would be $T_2 = |\overline{A_1D_1}| + |\overline{A_1D_2}| + |\overline{A_1D_3}| + |\overline{A_2D_4}| + |\overline{A_2D_5}|$, and the "progress" made is $T_1 - T_2$. The "cost" is the number of selected neighbors (i.e. 2 in the previous figure). Thus, the forwarding set $\{A_1, A_2\}$ is evaluated as $2/(T_1 - T_2)$. So, among all candidate forwarding sets, the one with optimal value of this expression is selected. Those destinations for which there is no neighbor closer to them, will be reached using GFG routing [26] directly to them. Note that the number of expressions to evaluate grows with number of neighbors and number of destinations. We provide below an enhanced algorithm to reduce the number of evaluations.

#### 4.2.1 Exhaustive Enumeration by Set Partitioning

Given $k$ destinations, the algorithm can consider all $S_k$ partitions. For each set given in the set partition, check whether there is a node which is closer to the destinations in the set than the current node $C$. If it is not possible to find such a node for a set, that particular partition is ignored. For those being possible, the cost/progress ratio is computed, and the best one is selected. This solution is applicable for small number of destinations (e.g. up to 5). For a larger number, it becomes exponential in $k$, and therefore a faster greedy solution is needed. A fast algorithm for generating set partitions is given in [30].

#### 4.2.2 Greedy Selection of Set Partitions

The goal of this greedy selection of set partitions is to reduce the number of partitions (destination sets) being evaluated. The destinations for which there is no closer neighbor are served using the GFG protocol [26]. Thus, we start with the set of destinations $\{D_1, D_2, \ldots, D_k\}$, for which there is at least one node closer to them than $C$.

For each of these $D_i$ we choose the best neighbor $A_i$ of $C$ as if it was the only destination. If there are several destinations for which the best neighbor is the same, then we merge those destinations into a single set $\{M_i\}$. At the end of this phase, we have an initial set partition of the destinations $\{M_1, M_2, ...M_l\}$, and there is a different neighbor $A_i$ for each subset $M_i$. Each $M_i$ has its cost over progress $1/P_i$, and the whole partition also has its overall cost over progress being equal to $l/(P_1 + P_2 + \cdots + P_l)$. In the example of Fig. 56.6, there are two subsets $M_1 = \{D_1, D_2, D3\}$ and $M_2 = \{D_4, D_5\}$. The cost over progress of $M_1$ is $1/P_1$ being $P_1 = (|\overline{CD_1}| + |\overline{CD_2}| + |\overline{CD_3}|) - (|\overline{A_1D_1}| + |\overline{A_1D_2}| + |\overline{A_1D_3}|)$. Similarly, the cost over progress for $M_2$ is $1/P_2$ being $P_2 = (|\overline{CD_4}| + |\overline{CD_5}|) - (|\overline{A_2D_4}| + |\overline{A_2D_5}|)$. The overall cost over progress ratio of the whole partition is then $2/(P_1 + P_2)$.

After this first iteration, we will repeatedly try to improve the cost over progress ratio, until this was not possible in a given iteration. In each iteration, we try to merge each pair $M_i, M_j$ to see whether they can improve the cost-progress ratio by merging into one set. This merge is done selecting a new $A_j$, being the neighbor of $C$ which is closer than $C$ to all destinations in $M_i \cup M_j$ and provides best ratio. If such $A_j$ does not exist, merge is not possible. From all possible merges which improve the ratio compared to the partition set, select the one with higher ratio, and update the set partition by effectively merging those sets, and decreasing $l$ by 1. The next iteration starts with this new set partition. The process is repeated until no new merging improves the ratio.

It is easy to prove that this algorithm would test $O(k^3)$ cases rather than $k!$.

The described algorithm is based on position information. If it is not available, it can be applied on a version where distances between nodes are replaced by hop counts between them. Each receiver may flood the network, so that each node may learn hop count distances to all receivers. These distances can be used in the described protocol to determine multicast routes.

# 5 Discussion

Multicast routing, has proven to be an interesting problem requiring approximation algorithms, and simple yet efficient heuristic solutions. As we have seen, multicast routing problems are NP-complete even when the metrics to optimize are not very complicated.

In this chapter, we have shown example of such greedy algorithms and their applicability to the multicast routing problem in two different environments: wireless ad hoc networks, and wireless sensor networks.

In ad hoc networks, we have shown that the traditional Steiner tree problem does not guarantee optimality regarding the overall bandwidth consumption of the multicast tree, and we have applied a heuristic epidemic algorithm to approximate efficiently those bandwidth-efficient trees.

For sensor network scenarios, we have shown how the general geographic routing concept can be extended to solve the multicast routing task in a localized way. In addition, we have explained how the general cost-progress ratio framework can be also used for such problem. As we showed, using this framework we avoid adding extra parameters to the problem formulation, which is one of the issues present in most of the existing solutions in the literature.

We believe that the concepts presented here will be useful for solving a number of other network layer problems, or even variants of these ones based on different assumptions or optimality criteria.

# References

[1] Deering, S., Multicast Routing in a Datagram Internetwork, *Ph.D. Thesis, Electrical Engineering Dept., Stanford University*, Dec. 1991.

[2] Deering S., and Cheriton, D.R., Multicast Routing in datagram internetworks and extended LANs, *Transactions on Computer Systems,* vol.8, no.2, May 1990, pp. 85–110.

[3] Moy, J., Multicast routing extensions for OSPF, *Computer communications of the ACM,* vol.37, no.8, August 1994, pp.61–66.

[4] Ballardie, T., Francis P., and Crowcroft, J., Core Based Trees (CBT) – An architecture for scalable inter-domain multicast routing, *Proc. of ACM SIGCOMM'93*, San Francisco, CA, October 1993, pp.85–95.

[5] Deering, S, Estrin, D., Farinacci, D., Jacobson, V., Liu, C.G., and Wei, L., The PIM architecture for wide-area multicast routing, *IEEE/ACM Transactions on Networking,* vol.4, no.2, April 1996, pp. 153–162.

[6] Wu, C.W., Tay Y.C., and Toh, C.K., Ad Hoc Multicast Routing Protocol Utilizing Increasing id-numberS (AMRIS) Functional Specification, Internet-draft, work in progress, draft-ietf-manet-amris-spec-00.txt, November 1998.

[7] Belding-Royer E.M, and Perkins, C.E. Multicast operation of the ad-hoc on-demand distance vector routing protocol, *Proceedings of ACM/IEEE MOBICOM'99,* Seattle, WA, August 1999, pp. 207–218.

[8] Ji L. and Corson, S., A Lightweight Adaptive Multicast Algorithm, *Proceedings of IEEE GLOBE-COM'98,* Sydney, Australia, November 1998, pp. 1036–1042.

[9] Jetcheva, J.G., Johnson, D.B., Adaptive Demand-Driven Multicast Routing in Multi-Hop Wireless Ad Hoc Networks, *Proceedings of ACM MobiHoc'01*, Long Beach, CA, October, 2001, pp. 33–44.

[10] Lee, S.J., Su, W., and Gerla, M., On-demand multicast routing protocol in multihop wireless mobile networks, *ACM/Kluwer Mobile Networks and Applications,* vol. 7, no. 6, pp. 441–452, December 2002.

[11] Garcia-Luna-Aceves, J.J., and Madruga, E.L., The Core-Assisted Mesh Protocol, *IEEE Journal on Selected Areas in Communications*, vol.17, no.8, August 1999, pp. 1380–1394.

[12] Toh, C.K., Guichala, G., Bunchua, S., ABAM: On-Demand Associativity-Based Multicast Routing for Mobile Ad hoc Networks, *Proceedings of IEEE VTC-2000*, Boston, MA,September 2000, pp.987–993.

[13] Bommaiah, E., Liu, M., MacAuley A., and Talpade, R., AMRoute: Ad hoc Multicast Routing Protocol, Internet-draft, work in progress, draft-talpade-manet-amroute-00.txt, August 1998.

[14] Sinha, P., Sivakumar, R., and Bharghavan, V., MCEDAR: Multicast Core-Extraction Distributed Ad hoc Routing, *Proc. of IEEE Wireless Commun. and Networking Conf.*, New Orleans, LA, September 1999, pp.1313–1319.

[15] Ji, L., and Corson, M.S., Differential Destination Multicast: A MANET Multicast Routing Protocol of Small Groups, *Proc. of IEEE INFOCOM'01*, Anchorage, Alaska, April 2001, pp. 1192-1202.

[16] Karp, R.M., Reducibility among combinatorial problems, *In Complexity of computer computations,* Plenum Press, New York, 1975, pp.85–103.

[17] Waxman, B.M., Routing of Multipoint Connections, *IEEE Journal on Selected Areas in Communications,* vol. 6, no. 9, December 1998, pp. 1617–1622.

[18] Kou, L., Markowsky, G., and Berman L., A fast algorithm for Steiner trees, *Acta Informatica,* no. 15, vol. 2, 1981, pp. 141–145.

[19] Plesnik, J., The complexity of designing a network with minimum diameter, *Networks,* no. 11, 1981, pp. 77–85.

[20] Zelikovsky A., An 11/6-approximation algorithm for the network Steiner problem, *Algorithmica,* no. 9, 1993, pp.463–470.

[21] Rajagopalan, S., and Vazirani, V.V., On the bidirected cut relaxation for the metric Steiner tree problem, *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms,* 1999, pp. 742–751.

[22] Clark, B.N., Colbourn, C.J., Johnson, D.S., Unit Disk Graphs, *Discrete Math.* No. 86, pp. 165- 177, 1990

[23] Ruiz, P.M. and Gomez-Skarmeta A.F., Mobility-Aware Mesh Construction Algorithm for Low Data-Overhead Multicast Ad hoc Routing, *Journal of Communications and Networks,* Vol. 6, no. 4, December 2004, pp. 331-342.

[24] Ruiz, P.M. and Gomez-Skarmeta A.F., Approximating Optimal Multicast Trees in Wireless Multihop Networks, *in proc. 10th IEEE Symposium on Computers and Communications, ISCC 2005,* La Manga, Spain, June 2005, pp. 686–691.

[25] Giordano, S., and Stojmenovic, I., Position Based Routing in Ad hoc Networks, a Taxonomy, *Ad How Wireless Networking*, X. Cheng, X. Huang and D.Z. Du (eds.), Kluwer, 2003.

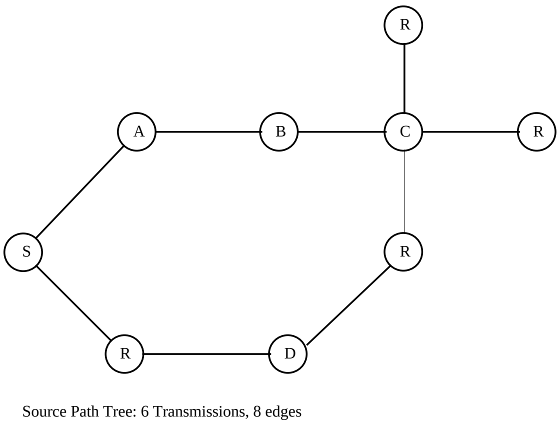
[26] Bose, P., Morin, P., Stojmenovic, I., and Urrutia, J., Routing with guaranteed delivery in ad hoc wireless networks, *ACM Wireless Networks,* Vol. 7, no. 6, November 2001, pp. 609–616

[27] Stojmenovic, I., and Lin, X., Power aware localized routing in wireless networks, *IEEE Transactions on Parallel and Distributed Systems,* Vol. 12, No. 11, November 2001, pp.1122–1133.

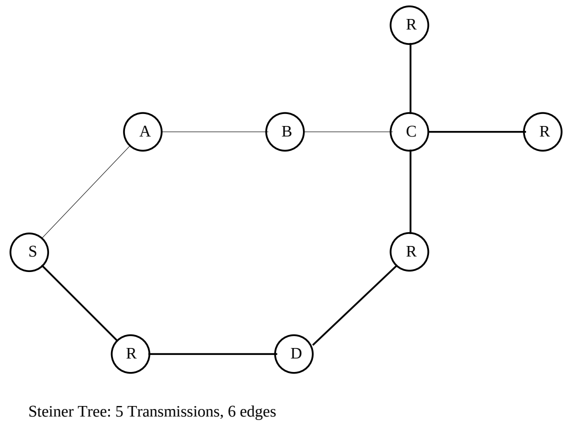
[28] Stojmenovic, I., Localized network layer protocols in sensor networks based on optimizing cost over progress ratio and avoiding parameters, Submitted for publication.

[29] Mauve, M., F*üβ*ler, H., Widmer, J., Lang, T., Position-Based Multicast Routing for Mobile Ad-Hoc Networks, TR-03-004, Department of Computer Science, University of Mannheim, March, 2003.

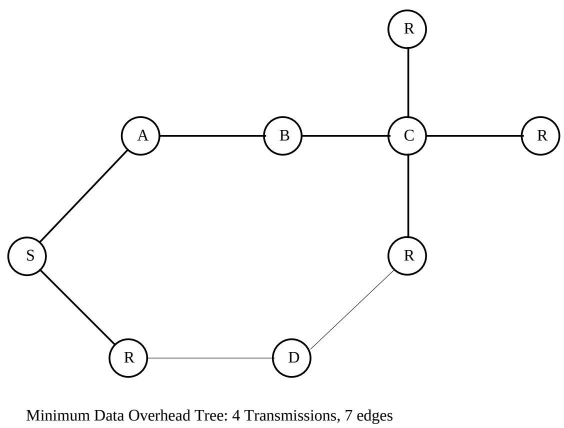
[30] Djokic B., Miyakawa M., Sekiguchi S., Semba I., Stojmenovic I., A fast iterative algorithm for generating set partitions, *The Computer Journal,* Vol. 32, No. 3, 1989, 281-282.




(a) Shortest Path Tree




(b) Steiner Tree




(c) Min. Data Overhead

Figure 56.1: Differences in cost for several multicast trees over the same ad hoc network

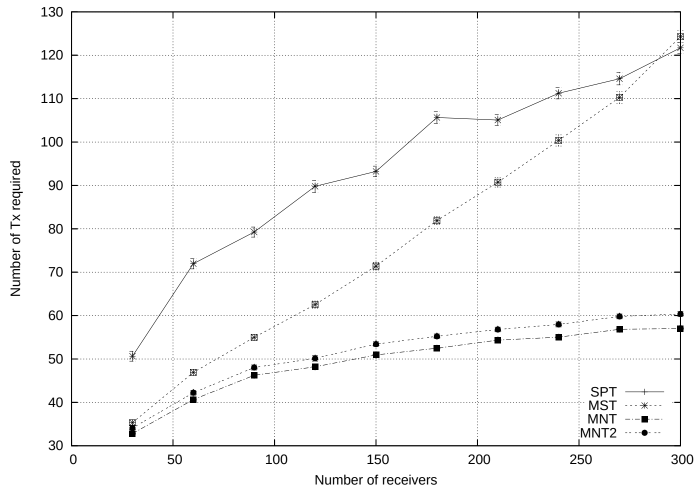



Figure 56.2: Number of Tx at increasing number of receivers.

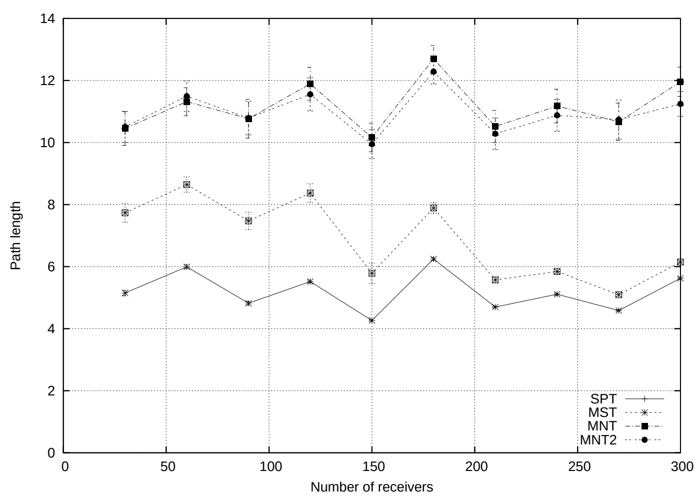



Figure 56.3: Mean path length at increasing number of receivers.

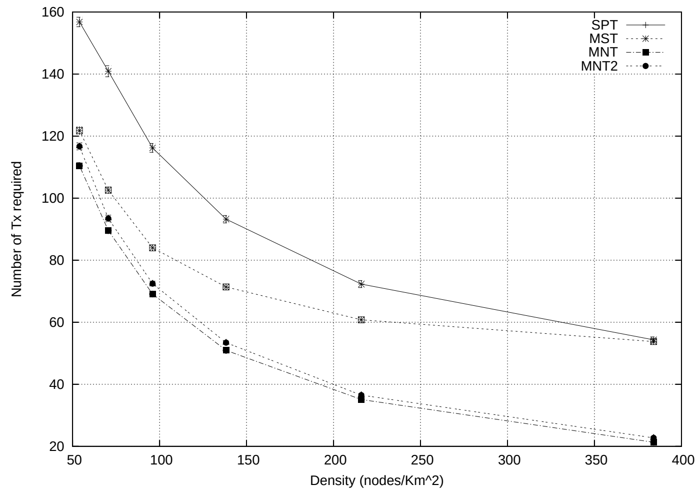



Figure 56.4: Number of Tx with varying network density for 150 receivers

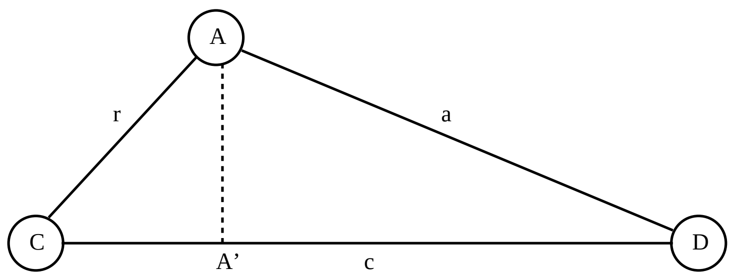



Figure 56.5: Best neighbor selection in localized routing schemes

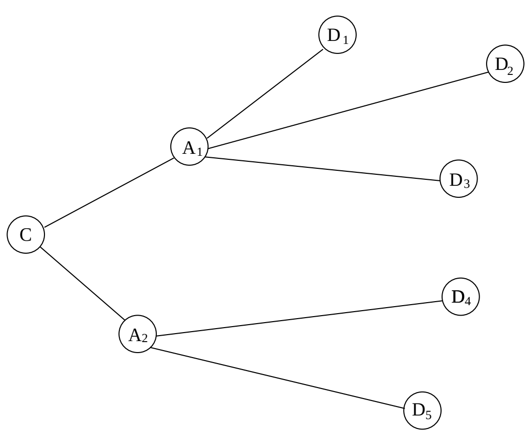



Figure 56.6: Evaluating candidate forwarding set from neighbors

# Index