Approximating Optimal Multicast Trees in Wireless Multihop Networks

Pedro M. Ruiz
Dept. Information and Communications Engineering
University of Murcia
Fac. Informatica, Campus de Espinardo s/n.
E-30071 Espinardo, Murcia (Spain)
E-mail: pedrom@dif.um.es

Antonio F. Gomez-Skarmeta
Dept. Information and Communications Engineering
University of Murcia
E-30001, Aptdo. 4021, Murcia (Spain)
E-mail: skarmeta@dif.um.es

***Abstract*— We study the problem of computing minimal cost multicast trees in multi-hop wireless mesh networks. This problem is known as the Steiner tree problem, and it has been widely studied in fixed networks. However, we show in this paper that in multi-hop wireless mesh networks, a Steiner tree is no longer offereing the lowest bandwidth consumption. So, we reformulate the problem in terms of minimizing the numbrer of transmissions. We show that the new problem is also NP-complete and propose heuristics to approximate such trees. Or simulations results show that the proposed heuristics offer a lower cost than Steiner trees over a variety of scenarios.**

I. Introduction and Motivation

A wireless multihop network consists of a set of nodes which are equipped with wireless interfaces. Nodes which are not able to communicate directly, use multihop paths using other intermediate nodes in the network as relays. When the nodes are free to move, these networks are usually known as ”mobile ad hoc networks”. We focus on this paper in static multihop wireless networks, also known as ”mesh networks”. These networks have recently received a lot of attention in the research community, and they are also gaining momentum as a cheap and easy way for mobile operators to expand their coverage and quickly react to temporary demands.

In addition, IP multicast is one of the areas which are expected to play a key role in future mobile and wireless scenarios. Key to this is the fact that many of the future services that operators and service providers forsee are bandwidth-avid, and they are strongly based on many-to-many interactions. These services require an efficient underlying support of multicast communications when deployed over multihop extensions where bandwidth may become a scarce resource.

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 might think 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.

The problem of finding a minimum cost multicast tree is well-known as the minimum Steiner tree problem. Karp [7] demonstrated that this problem is NP-complete even when every link has the same cost, by a transformation from the exact cover by 3-sets. There are some heuristic algorithms [8] to compute minimal Steiner trees. For instance, the MST algorithm ([9], [10]) provides a 2-approximation, and Zelikovsky [11] proposed an algorithm which obtains a 11/6-approximation. However, 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.

Similarly, the multicast ad hoc routing protocols proposed in the literature [6] do not approximate a minimal cost multicast tree either. For ad hoc networks, most of the works in the literature devoted to the improvement of multipoint forwarding efficiency have been related to the particular case of flooding (i.e. the broadcast storm problem). Only a few papers like Lim and Kim [13] analyzed the problem of minimal multicast trees in ad hoc networks, but they defined several heuristics based on the minimum connected dominating set (MCDS) which are only valid for flooding.

Although it is widely assumed that a Steiner tree is the minimal cost multicast tree, we show in this paper that it is not generally true in wireless multihop networks (see Fig. 1. The problem of minimizing the cost 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.

In this paper we show that the Steiner tree does not always give an optimal solution. Additional contributions of this papers are the demonstration that the problem of minimizing the cost of a multicast tree in a wireless mesh network is also NP-complete, and the proposal of enhanced heuristics to approximate such optimal trees, which we call minimal data overhead trees. Our simulation results show that the proposed heuristics produce multicast trees with a lower cost than the MST heuristic ([9]) for Steiner trees over a variety of scenarios. In addition, they offer a huge reduction in the cost compared to the shortest path trees used by most of the ad hoc multicast routing protocols proposed so far.

The remainder of the paper is organized as follows: section II describes our network model, formulates the problem and shows that it is NP-complete. The description of the proposed algorithm is given in section III. In section IV we explain our simulation results. Finally, section V provides some discussion and conclusions.

II. Network Model and Problem Formulation

A. Network model

We represent the ad hoc network as an undirected graph $G(V, E)$ where V is the set of vertices and E is the set of edges. We assume that the network is two dimensional (every node $v \in V$ is embedded in the plane) and mobile 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 iif $dist(v_1, v_2) \leq r$ (i.e. v_1 and v_2 are able to communicate directly). In wireless mobile ad hoc networks some links may be unidirectional due to different transmission ranges. However, given that lower layers can detect and hide those unidirectional links to the network layer, we only consider bidirectional links. That is, $(v_1, v_2) \in E$ iif $(v_2, v_1) \in E$.

B. Problem formulation

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 total number of transmissions required to deliver a packet from s to every receiver. To formulate the problem, we will need some previous definitions.

Definition 1. 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 can define a function $C_t : T \rightarrow \mathbb{Z}^+$ 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 1. Given a tree $t \in T$ as defined above, if we define the set F_t as the relay nodes in t, 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 lemma 1, the to minimize $C_t(t)$ we must somehow reduce the number of forwarding nodes $|F_t|$.

Definition 2. Under the conditions of definition 1, let $t^* \in T$ be the multicast tree such that $C_t(t^*) \leq C_t(t)$ for any possible $t \in T, t \neq t^*$. We define the data overhead of a tree $tinT$, 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 $T^* \subset G$ such that the following conditions are satisfied:

1) $T^* \supseteq V'$
2) $C_t(T^*)$ is minimum

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 say that $\omega_d(T^*)$ is minimum, and establishes the optimality of the tree. As we show in the next theorem, this problem is NP-complete.

Theorem 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

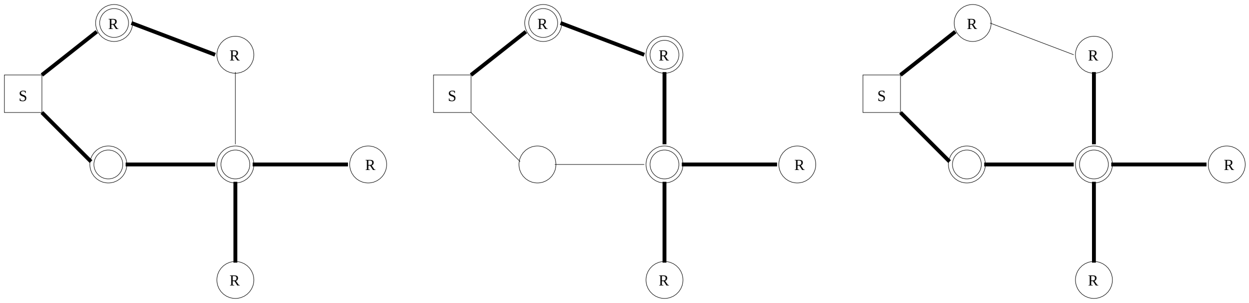
Proof: According to lemma 1, minimizing $C_t(T^*)$ is equivalent to minimize the number of relay nodes $F \subseteq T^*$. So, the problem is finding the smallest set of forwarding nodes F that connects s to every $r \in R$. If we consider the particular case in which $R = V - \{s\}$, the goal is finding the smallest $F \subseteq T^*$ which covers the rest of nodes in the graph $(V - \{s\})$. This problem is the well-known vertex cover problem [12], which is NP-complete. So, by including a particular case which is NP-complete, our problem is also NP-complete. ■

In the next theorem we show that in general the tree with the minimal edge-cost is not the one with the minimal data-overhead. Before presenting the theorem we give some definitions used within the proof of the theorem.

Definition 3. Under the same conditions of definition 1, and provided that each edge $e \in E$ has an associated cost $w(e) > 0$, we can define a function $C_e : T \rightarrow \mathbb{Z}^+$ so that given a tree $t \in T$, $C_e(t)$ is the edge cost of t defined as:

$$C_e(t) = \sum_{e \in E} w(e) \quad (1)$$

For the particular case of ad hoc networks, we can consider every edge to have the same cost. For simplicity in the calculations we assume that $w(e) = 1, \forall e \in E$. Even in that particular case, the problem of finding the multicast tree T^*



(a) Shortest Path Tree, 4 Tx/packet (b) Steiner Tree, 4 Tx/packet (c) Min. Data Overhead, 3 Tx/packet

Fig. 1. Differences in cost for several multicast trees over the same ad hoc network

so that $C_e(T^*)$ is minimum (also called Steiner tree) is NP-complete as R. Karp showed in [7]. In this particular case of unitary edge cost, $C_e(T^*)$ equals to the number of edges, which is $|V| - 1$ by a definition of tree.

Theorem 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: To proof the theorem we will show that given an Steiner tree T^*, it is possible to find a tree $T^{'}$ such that $C_e(T^{'}) \geq C_e(T^*)$ and $C_t(T^{'}) \leq C_t(T^*)$. Let's denote by $F^{'}$ and F^* the number of forwarding nodes in each of the trees. For $T^{'}$ to offer a lower data overhead, the following condition must hold:

$$1 + F^{'} \leq 1 + F^*$$

In a multicast tree, the number of forwarding nodes can be divided into those which are also receivers and those who are not. The latter are usually called Steiner nodes, and we will denote the set of such nodes as $\mathbb{S}$. The number of forwarding nodes which are receivers can be easily computed as $|R - \mathbb{L}|$ being $\mathbb{L}$ the set of leaf nodes. Of course, every leaf node is also a receiver. Thus, the previous inequality, is equivalent to the following one:

$$\begin{aligned} 1 + |\mathbb{S}^{'}| + (|R| - |\mathbb{L}^{'}|) \leq 1 + |\mathbb{S}^*| + (|R| - |\mathbb{L}^*|) \Rightarrow \\ |\mathbb{S}^{'}| - |\mathbb{L}^{'}| \leq |\mathbb{S}^*| - |\mathbb{L}^*| \Rightarrow \\ |\mathbb{S}^{'}| - |\mathbb{S}^*| \leq |\mathbb{L}^{'}| - |\mathbb{L}^*| \end{aligned} \tag{2}$$

In addition, by the definition of C_e, $C_e(T^{'}) \geq C_e(T^*) \Rightarrow |V^{'}| - 1 \geq |V^*| - 1$. Given that the number of vertex is exactly the sender plus the number of Steiner nodes plus the number of receivers, we can derive the following inequality:

$$|\mathbb{S}^{'}| + |R| \geq |\mathbb{S}^*| + |R| \Rightarrow |\mathbb{S}^{'}| \geq |\mathbb{S}^*|$$

So, according to Eq. 2 it is possible to build a tree $T^{'}$ so that $C_t(T^{'}) \leq C_t(T^*)$ provided that the number of additional steiner nodes added ($|\mathbb{S}^{'}| - |\mathbb{S}^*|$), is lower than the additional number of leaf nodes ($|\mathbb{L}^{'}| - |\mathbb{L}^*|$). ■

As expected, this means that the Steiner tree reduces the cost by minimizing the number of Steiner nodes $|\mathbb{S}^*|$. This is equivalent to say that it tries to maximize the number of receivers which are included in F^*, which in turn reduces the number of leaf nodes $|\mathbb{L}^*|$. An example of such a tree is shown in Fig.1.

III. Proposed Algorithms

Given the NP-completeness of the problem, within the next subsections we describe two heuristic algorithms 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.

A. 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 2, this algorithm 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. The algorithm shown in algorithm 1, starts by removing 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 more 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 Steiner nodes, guaranteeing the cost-effectiveness. In the latter case, the different subtrees are connected by an 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

MF $\leftarrow \oslash / * mcastforwarders * /$
V $\leftarrow$ V - $\{s\}$
aux $\leftarrow$ R-Cov(s) + $\{s\} / * nodestocover * /$
repeat
  node $\leftarrow argmax_{v \in V}(|Cov(v)|)$ 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 using MST heuristic
end if

Theorem 3. The proposed algorithm results in a tree with a lower or equal data-overhead than the one resulting from the MST Steiner tree.

Proof: Let's consider the worse case in which no cost-effective tree can be formed. There are two possible cases:

1) There are no receivers in the range of the source. Then Cov(s)=$\oslash$ and the resulting tree (T_1) is exactly a Steiner tree among s and all the receivers computed using the MST heuristic (T_2). Thus, $C_t(T_1) = C_t(T_2)$.
2) There are receivers in the range of the source. Then, the resulting tree (T_1) is a Steiner tree among the source s and all the receivers except Cov(s) computed with the MST heuristic. This tree is a subtree of the Steiner tree from s to every receiver (T_2), so $C_t(T_1) \leq C_t(T_2)$.

■

B. Distributed approximation algorithm

Centralized algorithm may be useful for some kind of networks, however a distributed approach is much more appealing for the vast majority of scenarios. In this section we present a slightly different version of the previous algorithm, being able to be run in a distributed way.

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, the previous protocol used the MST heuristic. However, this is a centralized heuristic consisting of two different phases. Firstly, the algorithm builds the metric closure 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 tree between the to nodes connected by that edge. Unfortunately, the metric closure of a graph is hard to build in a distributed way. However, we can approximate such an 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. The way in which the algorithm is executed from the source-root to the other nodes guarantees that the obtained tree is connected.

Algorithm 2 Distributed approximation of MST heuristic

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

The second part of the algorithm to make distributed is the creation of the cost-effective subtrees. However, this part is much simpler and 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 which covers. This information is known locally by just 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 a higher 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.

In the next section, we shall see that this distributed version of the algorithm offers is not as efficient as the centralized one, but offers a good approximation to the centralized scheme. This is because instead of really computing the metric closure in the graph, we just approximate it. However, the performance of the distributed approach is still better than the one offered by the Steiner tree.

IV. Simulation Results

In order to assess the effectiveness of our proposed algorithms we have simulated them under different conditions. The algorithms that we have simulated are the two proposed approaches as well as the MST heuristic to approximate Steiner trees. In addition, we also simulated the shortest path tree algorithm, which is the one which is used by most multihop multicast routing protocols proposed to date.

A. Performance metrics

We are interested in evaluating the optimality of the the topology of the multicast tree produced by the different algorithms. That is the reason why we use different metrics from the typical performance measurements (e.g. packet delivery ratio) which strongly depend upon the underlying wireless technology under consideration. In our particular case, the metrics under consideration are:

- Number of transmissions required. The total number of packet transmitted either by the source or by any relay node to deliver a data packet from the source to all the receivers.
- Mean number of hops. The number of multicast hops from a receiver to the source averaged over the total number of receivers.

So, by considering these metrics along with a perfect MAC layer (i.e. without collisions, retransmissions or interferences) we guarantee an unbiased comparison.

B. Simulation methodology

All the approaches have been evaluated under a different number of receivers, and a varying density of the nodes in the network. In particular, the number of receivers considered was between 1 and 40% of the nodes, which corresponds to the range from 5 to 200 receivers. The density of the network varied between 100 and 500 $nodes/Km^2$.

For each combination of simulation parameters, a total of 91 simulation runs with different randomly-generated graphs were performed, making a total of more than 100000 simulations. The error in the graphs shown below are obtained using a 95% confidence level.

C. Performance evaluation

In the figures below, SPT refers to the shortest path tree and MST corresponds to the MST heuristic to approximate Steiner trees. Finally, MNT and MNT2 correspond to the proposed centralized and distributed heuristics respectively.

In Fig. 2 we show for a network with an intermediate density how the number of transmissions required varies with respect to the number of receivers. As expected, when the number of receivers is lower than 20, the proposed schemes do not offer significative differences compared to the Steiner tree heuristic. This is clearly explained by the fact that the nodes tend to be very sparse and it is less likely that it is possible to build cost-effective trees. However, as the number of receivers increases, the creation of cost-effective trees is favored, making the the proposed schemes to achieve significative 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. Regarding the two proposed approaches the distributed approach, by avoiding the use of the metric closure, gets a slightly lower performance compared to the centralized approach. However, both of them have a very similar performance which allow them to offer substantial bandwidth savings compared to the Steiner tree (i.e. MST heuristic).

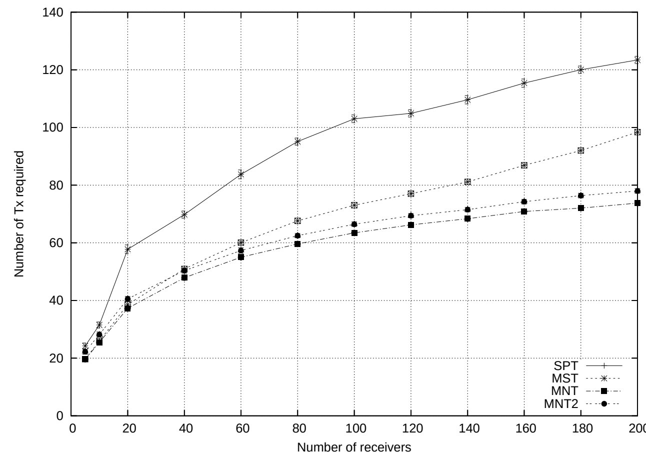


Fig. 2. Total number of transmissions at increasing number of receivers.

To evaluate the impact on the length of the paths, we performed the analysis shown in Fig. 3. As expected the SPT the one offering the lowest mean path length. The Steiner tree heuristics as well as the proposed ones offer a higher mean path length. This is clearly due to the fact grouping paths for several receivers makes them not to use their shortest paths. As we can see, the 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.

Another important aspect to consider is how the performance varies regarding the density of the network. This results are of paramount importance to determine under which scenarios the proposed approaches behave better. In particular we consider two different cases: a medium number of receivers represented by a 20% of the nodes, and a high number of receivers represented by a 36% of the nodes. As we show before, the case of a very low number of receivers is no interesting because most of the approaches offer a similar performance.

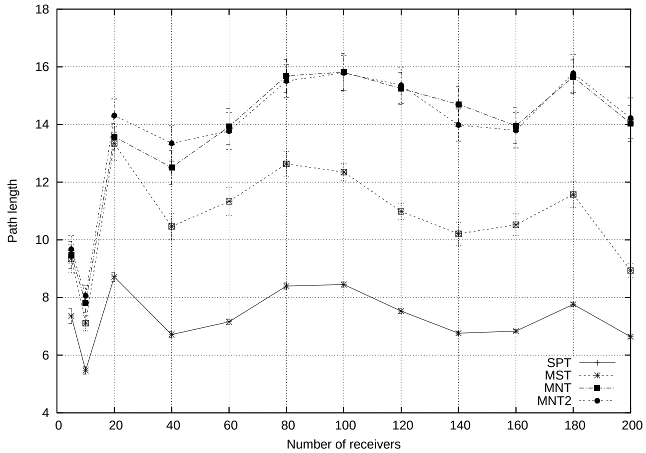


Fig. 3. Mean path length at increasing number of receivers.

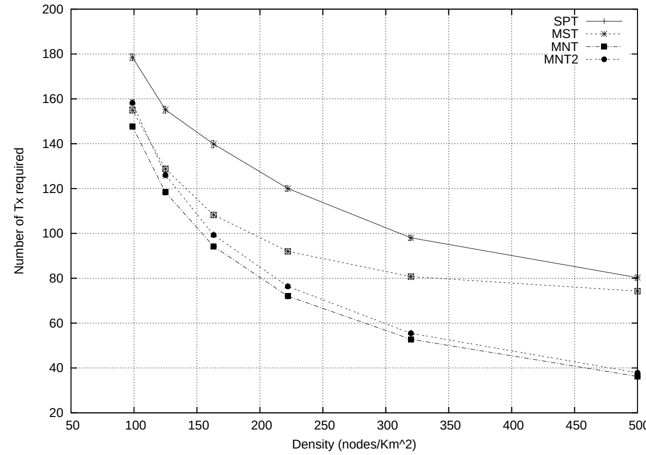


Fig. 5. Number of Tx for 180 receivers with varying network density.

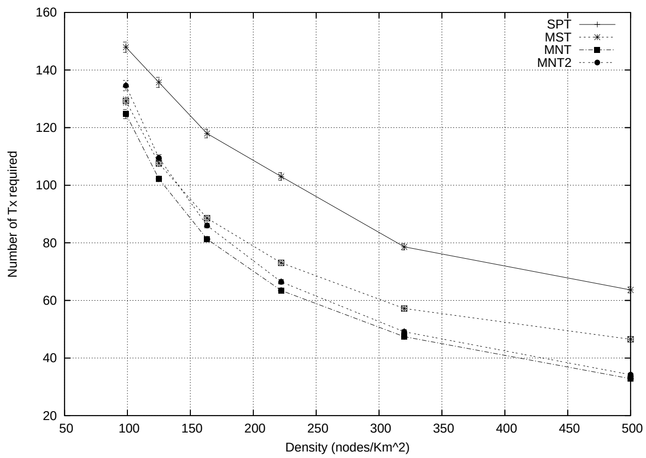


Fig. 4. Number of Tx for 100 receivers with varying network density.

In Fig. 4 and Fig. 5 we present the results for the medium number of receivers and high number of receivers respectively. As the figure depicts, the higher the density, the better is the performance of all the approaches. This makes sense, because the higher the density the lower the path lengths, 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 that our proposed heuristics achieve compare to the other approaches is higher as the density of the network 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.

In addition, the higher the density, the closer in performance are the centralized and the distributed approaches. This is because in dense networks, the number of hops between any pair of nodes is also reduced. This makes the difference between metric closure and its approximation in number of hops to be reduced as well. This makes our approach very appealing for dense networks such as sensor networks in which the mean degree of a node is usually very high.

If we compare the two figures we can see that the difference in the number of receivers just varies a little bit the concrete performance differences among approaches. However, the density of the network has an strong effect on the overall performance of the solutions.

V. Conclusions and discussion

As we have shown, the generally considered minimal cost multicast tree (Steiner tree) does not offer an optimal solution in multihop wireless networks. The problem is that the original Steiner tree problem formulation does not account for the reduction in bandwidth that can be achieved in a broadcast medium. Given those limitations we re-formulate the problem in terms of minimizing the number of transmissions required to send a packet from a multicast source to all the receivers in the group.

We have shown that this formulation is adequate for multihop wireless networks, and we have also demonstrated that this problem is NP-complete. So, we have introduced two new heuristic algorithms to deal with the problem of optimizing multicast trees in wireless mesh networks. Our simulation results show that the proposed heuristics manage to beat the Steiner tree MST heuristic over a variety of scenarios and network densities.

In particular, our results show that the higher the density of the network, the higher are the performance gains introduced by our heuristics compared to the other approaches. These results seem very promising as a possible future direction to address similar issues in sensor networks in which the network topology is generally very dense, and reverse multicast trees are very common as a mechanism to gather information from the sensor network.

Acknowledgment

Part of this work has been funded by Spanish MCYT by means of the ”Ramon y Cajal” workprogramme, the ICSI Call for Spanish Technologists and the SAM (MCYT, TIC2002-04531-C04-03) project. This work has been performed during a Visiting Fellowship of Pedro M. Ruiz at the International Computer Science Institute (www.icsi.berkeley.edu).

References

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

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

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

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

[6] C. Cordeiro, H. Gossain and D. Agrawal, ”Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions” *IEEE Network*, no. 1, Jan 2003, pp. 52–59. K.Bharath-Kumar and J.-M. Jaffe, ”Routing to Multiple Destinations in Computer Networks,” *IEEE Transactions on Communications,* no.31 Vol.3, 1983, pp. 343–351.

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

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

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

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

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

[12] S. Even, ”Graph Algorithms,” *Computer Science Press,* 1979, pp. 204–209.

[13] H. Lim and C. Kim, ”Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks,” *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, Boston, MA, USA. August, 2000, pp. 61–68.