

Improving Multicast Tree Construction in Static Ad Hoc Networks

Aleksi Penttinen and Jorma Virtamo

Networking Laboratory, Helsinki University of Technology

P.O. BOX 3000, FIN-02015 HUT, Finland

{Aleksi.Penttinen, Jorma.Virtamo}@hut.fi

Abstract

We consider wireless multicast tree construction for energy efficiency. We introduce a novel algorithm, Incremental Shortest Path Tree (ISPT), to generate source-based multicast trees and analyse its performance. Additionally, the potential of further improvements in the tree construction is addressed comparing the performance of the algorithm with the results from a simulated annealing optimisation.

1. Introduction

Ad hoc networks are typically deployed by a group of people (or vehicles, computers etc.) set out to complete a task in an environment where no existing network infrastructure is available. These tasks, such as emergency rescue operations, battlefield missions or shared desktop meetings, are likely to contain applications (walkie-talkie, live video streams, surveillance data) where considerable amount of data is delivered to several destinations at the same time.

In some settings, such as in sensor networks, the node energy resources are scarce and, additionally, expensive, difficult or even impossible to replenish. Given that the sensor nodes are fairly stationary and their transmission powers are adjustable, well designed multicast trees can significantly reduce the energy consumption and improve efficiency.

This paper focuses on the energy efficient multicast tree problem; problem of selecting a set of sequential transmissions which connect a sender to a set of receivers so that the sum of the transmission costs is minimised. The problem is closely related to the Steiner Tree problem of graphs, but with one crucial difference: the radio transmissions are omni-directional and all the nodes within the transmission range receive the transmitted information with only one power cost. Thus, the costs are associated to nodes instead of links.

The energy efficient multicast tree problem in ad hoc networks was introduced by Wieselthier et al. in a series of pa-

pers [4, 5, 6] considering energy efficient broadcasting and multicasting. They also proposed an algorithm, MIP (Multicast Incremental Power) [5], for the problem. MIP constructs the multicast tree as follows. First, a broadcast tree is constructed as in the well-known Prim's algorithm with the difference that each step considers finding the minimum *incremental cost* (see Section 2 for definition) that is needed to connect the next node to the evolving spanning tree. The broadcast tree is then pruned by eliminating all the transmissions that are not needed to reach the members of the multicast group.

We introduce a novel algorithm to generate source based multicast-trees and analyse its performance. The proposed algorithm, Incremental Shortest Path Tree (ISPT), starts with an initial tree and then grafts the receivers one by one to the tree using paths that yield the lowest incremental costs. It is shown that the algorithm is especially suitable in cases where the number of multicast receivers is fairly small. Using an energy-aware tree construction method with a suitable higher layer algorithm to share the traffic between trees, it is also possible to explicitly maximise the lifetime of a network that uses multicast communications [1, 3].

The organisation of this paper is as follows. Section 2 describes the network model and Section 3 presents our proposed Incremental Shortest Path Tree algorithm. In Section 4 ISPT is compared with MIP in a simulation study, while Section 5 applies simulated annealing to the problem to obtain reference results. Finally, we conclude in Section 6.

2. Network model

We consider a relatively static ad hoc network, such as a sensor network, with a set of identical nodes $i \in \mathcal{N}$ with omni-directional radio transmitter with adjustable transmission power (\sim range), possibly with an upper limit. Each node is assumed to know its own transmission cost (in terms of power or capacity usage etc.) to its neighbours. These costs constitute a link cost matrix of the network with the elements c_{ij} ($c_{ij} = \infty$ if no connection between i and j).

For all $i \in \mathcal{N}$ the *state* of node i is identified with its transmission cost. Possible states of node i are thus $s_i \in \{c_{ij} \mid c_{ij} \neq \infty\}$. When node i is not transmitting we set $s_i = 0$. In state $s_i = c_{ij}$ node i is transmitting with a power just enough to reach node j . We also use the term node cost interchangeably with the current state of the node. The set $S = \{s_i \mid i \in \mathcal{N}\}$ denotes the *current state of the network*.

Consider a node i in the tree which is currently in state s_i . The *incremental cost* to reach neighbour j , Δc_{ij} , is defined as the difference $\Delta c_{ij} = \max(c_{ij} - s_i, 0)$.

Given a state S of the network, the elements Δc_{ij} constitute the incremental link cost matrix. The *incremental path cost* is defined as the path cost calculated using the incremental link cost matrix of the network.

3. ISPT - Incremental Shortest Path Tree

ISPT starts from an initial tree containing the source and forms a tree iteratively by grafting the designated receiver nodes to the tree one by one.

3.1. Algorithm description

The initial tree, trunk, is defined here to be a subtree which originates from the multicast source. In our simulations, we have studied two variants of the ISPT algorithm, ISPT1 and ISPT2, which differ only in the selection of initial trees. In ISPT1 the source node is used as the trunk, whereas in ISPT2 it is the shortest path from the source to the “most distant” multicast receiver (for which the shortest path is the most expensive). These selections are rather arbitrary but found to produce good results.

Starting with initial tree as the current tree, the multicast tree is then constructed incrementally by repeating the following grafting step: For each receiver not yet in the current tree, determine the path from the tree to the receiver which yields the smallest incremental path cost. Having now one possible path for each receiver, select the path which has the smallest cost (again, an arbitrary but “well selected” choice for grafting order) and attach it to the tree to produce the current tree for the next iteration, see Figure 1 for illustration of ISPT2.

The shortest incremental paths can be found by applying the Dijkstra’s shortest path algorithm to a modified network. The modified network is the original network in which all the nodes of the current tree are merged into one node. The links from this node to its neighbours are determined according to the shortest incremental costs from the tree.

Less elegant, but more practical alternative is that the algorithm utilises only the original shortest path information. The information can be obtained from unicast routing tables or by running the shortest path algorithm only once for all pairs. The idea is that an incremental path cost is just

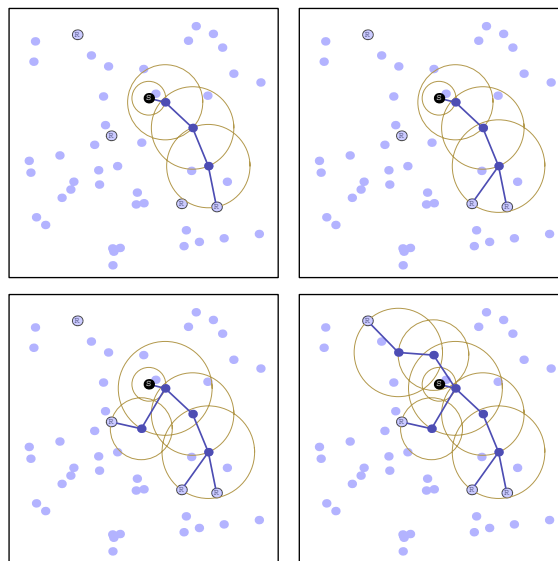


Figure 1. ISPT2 Example: Initial tree is the shortest path from the source to the furthest receiver. The grafting is repeated iteratively by attaching a path connecting the tree to a receiver, for which the incremental path cost is lowest. Algorithm stops after all the receivers are connected.

the sum of the incremental cost to a non-tree node, which is local information, and the ordinary shortest path cost from the non-tree node to the receiver. This requires only that the update information of the current best distances is relayed in the tree during its construction. The centralised operation of this implementation is described in Section 3.2.

The worst case complexity of the algorithm is $O(|\mathcal{N}|^3)$, where $|\mathcal{N}|$ is the number of nodes in the network. The worst case corresponds to the spanning tree problem in a fully connected network. However, the computational complexity is roughly the product of the number of nodes in the tree, average number of neighbouring nodes and number of receivers. Therefore, it is obvious that the smaller the tree the better ISPT performs in terms of computation time.

3.2. Summary of the algorithm

Denote the tree by T , the set of multicast receivers by D and the set of states (at what cost the node is transmitting to a neighbour) associated with the nodes by $S = \{s_i \mid i \in \mathcal{N}\}$ as defined in Section 2. Initially, set the tree to $T = \{s\}$, where s stands for the source node and set the destination distance list to $l_k = d_{sk}, \forall k \in D$, where d_{sk} is the ordinary shortest path distance between nodes s and k . Finally, set

the node state list $s_i = 0 \forall i$. The operation of ISPT is summarised in Algorithm 1.

Algorithm 1 ISPT operation

Tree initialisation: e.g. $T =$ shortest path to the receiver $\{k\} \max_k l_k$.

while there is a multicast receiver $k \notin T$ **do**
 grafting, see Algorithm 2
end while
 remove unnecessary transmissions by a sweep operation
 see Algorithm 3

Algorithm 2 Grafting algorithm

for each transmitting tree node i , for which the transmission cost has changed **do**
 for each neighbour j of i **do**
 for each destination k **do**
 if $\Delta c_{ij} + d_{jk} < l_k$ **then**
 set $l_k = \Delta c_{ij} + d_{jk}$
 associate l_k to (i, j, k)
 end if
 end for
 end for
 pick (i, j, k) for which l_k is smallest (or use selected grafting order)
 increase the state of node i to c_{ij}
 add the route $j \rightarrow k$ to the tree T, S .

Algorithm 3 Sweep operation

repeat
 list all the nodes \mathcal{N} in breadth-first-search from the source node
for each transmitting node $i \in T$ in the BFS-list **do**
 select $j = \arg \max_{j \in T} c_{ij}$ which does not yet have a predecessor
 set $s_i = c_{ij}$
 set i to be the predecessor of neighbours $\{k | c_{ik} \leq c_{ij}\}$
end for
until no changes in S (\sim two first iterations are enough)

4. Comparison of MIP and ISPT

In this section we compare MIP and ISPT in a large number of randomly generated test cases. In each case, we have used an area of 10x10 units in which the node coordinates have been drawn from uniform distribution. Multicast source and receivers have been also selected randomly

among the nodes. The number of nodes and receivers are varied.

The maximum transmission range is not limited but the form of the cost function assures that short links will be preferred. It should be noted, however, that a maximum value would only simplify the calculations as the number of neighbours for each node is smaller.

The link cost is taken to be $c_{ij} = r_{ij}^\alpha$, where r_{ij} is the distance between the nodes i and j and $2 \leq \alpha \leq 4$ as in [5].

Table 1 shows the mutual comparisons of the tested algorithms ISPT1, ISPT2 and MIP. \bar{x} stands for the mean *relative difference* and s for its estimated standard deviation. The relative difference statistics x for each problem sample is calculated as the ratio $x = \frac{alg1 - alg2}{alg1}$, where $alg1$ and $alg2$ are the results given by the first and the second algorithm in comparison, respectively. In other words, the statistics tells that the second algorithm in comparison yielded $x \times 100\%$ better result than the other algorithm.

ISPT provides a significant improvement in the tree cost in cases with small receiver groups while matching the MIP performance in larger ones. The cheaper the long distance transmissions, the better the performance of ISPT as it is able to identify good crossing points, where the multicast advantage can be effectively utilised. When the long distance transmissions are expensive, as in the case of $\alpha = 4$, the difference is smaller as the spanning tree paths (and short connections generally) become in relative terms more economical.

The performance of ISPT2 is explained by the good choice of initial tree. The long path as an initial tree provides more alternatives for the new branches, thus often avoiding the tree junctions concentrating near the source and finding better crossing points to utilise the advantage of omni-directional transmissions.

Although the average performance of ISPT2 seems to be better than that of the others, it does not guarantee the performance in any single case due to the heuristic nature of the algorithms. On the other hand, this brings out another major advantage of ISPT algorithm. It provides a simple way to produce a set of "reasonable" trees with significantly different costs by varying only initial tree or grafting order.

5. Simulated annealing

The well-known simulated annealing method [2] can be applied to the multicast problem to get reference results.

In the multicast problem we define the solutions as transmission patterns (a pattern of sequential transmissions starting from the source) that cover all the receivers. Solution neighbourhood is defined to consist of the patterns that are reached by an increase or decrease of a transmitter's power (state) to reach one new leaf/internal node or to drop a leaf

$\alpha = 2$	MIP-ISPT1		MIP-ISPT2		ISPT1-ISPT2	
Receivers	\bar{x}	s	\bar{x}	s	\bar{x}	s
5	0.194	0.135	0.196	0.136	-0.001	0.088
10	0.146	0.108	0.156	0.108	0.008	0.081
20	0.092	0.089	0.117	0.078	0.025	0.065
50	0.015	0.063	0.048	0.053	0.032	0.055
$\alpha = 4$						
5	0.112	0.128	0.110	0.135	-0.004	0.077
10	0.073	0.101	0.083	0.099	0.009	0.064
20	0.035	0.094	0.056	0.089	0.019	0.065
50	-0.021	0.075	0.014	0.059	0.032	0.058

Table 1. Comparison of the algorithms from 1000 samples of 100-node network instances

$\alpha = 2$	ISPT2-Optimisation		
Receivers	\bar{x}	s	Max. x
5	0.024	0.053	0.280
10	0.029	0.057	0.292

Table 2. Comparison of ISPT2 and simulated annealing results from 200 samples of 30-node network instances

node. The overall tree cost acts as the optimisation criterion as in the standard simulated annealing.

In this implementation, each tree problem was run 10^6 iteration steps and the solution is taken to be the best solution visited (instead of the stopping state). The temperature was decreased as 0.955^i where $i = 1 \dots 100$, and the process was run 10000 steps in each temperature. The annealing is started from the tree produced by ISPT2 (so the results are at least as good as ISPT2).

Table 2 shows the mean relative difference \bar{x} and its standard deviation s along with the maximum relative difference. The average tree cost seems to decrease slightly (as the simulated annealing always produced ISPT2 tree or a better one), especially when the receiver group grows. This seems natural as ISPT2 is optimal for unicast – the more junctions there are in the tree, the less optimal they become. However, the important observation here is that there are cases where ISPT2 does not perform well, as almost 30% improvements are possible.

6. Conclusions

This paper introduced ISPT, a novel multicast tree construction algorithm for ad hoc networks, which outperforms existing ones both in accuracy and simplicity when considering relatively small receiver groups (containing less than half of the nodes in the network). Furthermore, it is a simple

way to generate many different energy efficient multicast trees by different combinations of initial tree and grafting order.

We compared the results with those obtained using a simulated annealing optimisation. Although there are cases where the performance of ISPT falls short of optimal, the average behaviour seems to compete with the brute-force method of heuristic optimisation.

Acknowledgements

This work was funded by Finnish Defence Forces Technical Research Center and Graduate School in Electronics, Telecommunications and Automation (GETA).

References

- [1] J.-H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *INFOCOM*, volume 1, pages 22–31, 2000.
- [2] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimisation by simulated annealing. *Science*, 220(4589):671–680, 1983.
- [3] M. N. Vaello. Optimal routing of multicast traffic in ad hoc networks. M.Sc. thesis, Helsinki University of Technology, 2001.
- [4] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. Algorithms for energy-efficient multicasting in ad hoc wireless networks. In *Proceedings of IEEE MILCOM 1999*, volume 2, pages 1414–1418, 1999.
- [5] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proceedings of IEEE INFOCOM 2000*, pages 585–594, 2000.
- [6] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. Algorithms for energy-efficient multicasting in static ad hoc wireless networks. *ACM Mobile Networks and Applications*, 6:251–263, 2001.