α Tree in Sensor Network

Patrick Y.H. Cheung, and Nicholas F. Maxemchuk, Fellow, IEEE

Routing Problem in Sensor Network

We view the objective of the routing layer to be to choose paths through the sensor network to the sinks that maximize the lifetime of the network by minimizing energy consumption. Two competing properties of a sensor network must be considered when selecting paths:

(a) Data from different sensors can be aggregated together, reducing the overall energy costs. However, only data that traverses common paths can be aggregated together.

(b) The message arrival process is an impulse. When an event occurs, many sensors detect the event and report their measurements. During an impulse, those sensors that are chosen to act as forwarders will consume energy at a much higher rate. If the paths are not carefully provisioned, "popular" routes will run out of energy before the transmission of the impulse is complete.

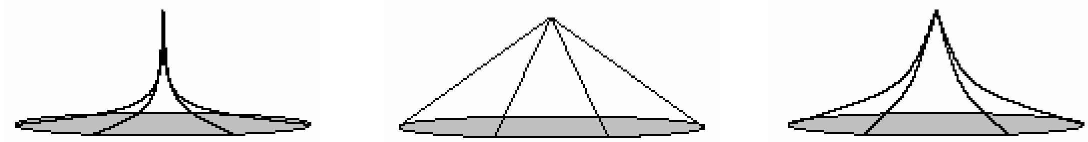
There are two competing effects. On the one hand concentrating the data on a small number of paths increases the compression and reduces the energy, while on the other hand concentrating the data on a smaller number of paths increases the energy expended by those nodes and decreases the network lifetime. Our objective is to determine the optimum concentration, taking both effects into account.

We will attack the problem in three phases. In the first phase, we minimize the total energy, taking into account the amount of aggregation that can be performed along the paths. In the second phase, we also consider the energy expended by the intermediate nodes. In the third phase, we take into account congestion and energy deficits and use deflection routing to move packets in directions that may be suboptimal on the average, but are preferable based on the actual use that the network has experienced. In the next section, we will describe the α tree algorithm, which is a response to the challenged proposed in the first phase.

α Tree Algorithm

When data is not aggregated or compressed, the routing structure that uses the least energy to transmit the data from the sensors to a destination is a minimum depth tree (MDT) that is rooted at the sink, with link weights indicating the energy needed to transmit on a link. In contrast, when the information from multiple sensors is completely redundant, only one unit of message is forwarded to the destination, regardless of the number of incoming messages. The tree that uses the least energy to collect the data in this case is a minimum spanning tree (MST). In most cases, the data from multiple sensors is not completely redundant, but has some redundancy. In the α tree algorithm, a single parameter α can be adjusted according to different levels of data redundancy in order to find routes that minimize the overall energy consumption.

In reference [1] we relate algorithms that generate minimum spanning trees and minimum depth trees by joining one node at a time to a tree that is rooted at the destination. In the minimum depth tree the nodes that are connected to the rooted tree are labeled with the distance to the destination. In the minimum spanning tree the node labels are zero. We plan on labeling the nodes with $\alpha \times$ (the distance to the destination) and relating α to the amount of data compression that is performed at the intermediate nodes.



(a) (b) (c)

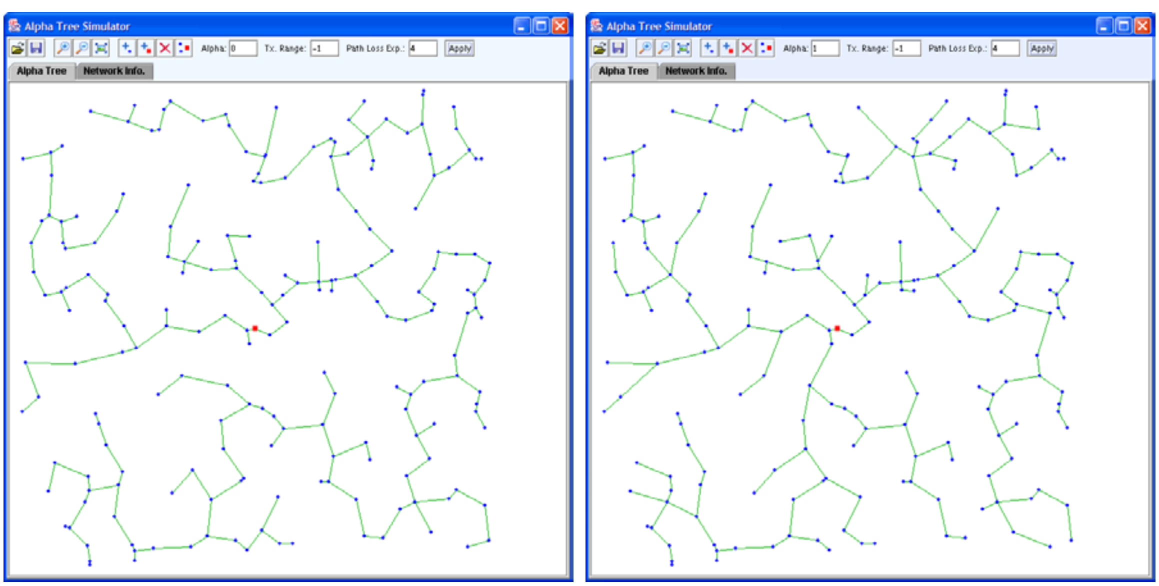
Figure 1 (a) Minimum Spanning Tree (b) Minimum Depth Tree (c) α Tree α trees, with $0 \le \alpha \le 1$ are between the minimum spanning and minimum depth trees. How α affects the "shape" of a tree is illustrated in Figure 1. As $\alpha \to 0$, we only consider the energy needed to connect the next node to the tree. The resulting tree is a minimum spanning tree, as depicted in Figure 1(a). When the data is compressed very little additional energy is required to get the rest of the way to the destination. As $\alpha \to 1$ equal weight is given to transmitting the data to the first node as along the rest of the path to the destination, which is true when there is no compression. The resulting tree is a minimum depth tree, as depicted in Figure 1(b). Intermediate values of α yield trees that fall somewhere in between the minimum spanning tree and minimum depth tree, as depicted in Figure 1(c). Our objective is to determine the α tree that best models the amount of data compression that can be performed.

In this initial description we have made α a constant. We can use an algorithm that is similar to Dykstra's shortest path algorithm to generate an α tree. We will also consider making α a function of the number of hops to the destination. When there is more than one sink for the data, we have a forest of trees. In order to find the forest of α trees we will start the algorithm with all of the destinations labeled as zero. Then, run the α tree algorithm for each sink successively. A node should switch its connection to a new tree whenever its distance to the new sink is shorter than its distance to the previous one.

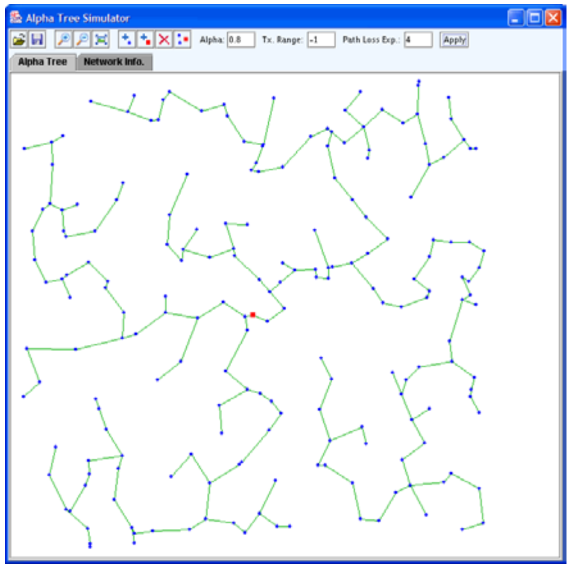
Preliminary Results and Impacts

To visualize the behavior of the α tree algorithm, an α tree simulator has been built. In our implementation, the link weights are set to be *(distance between two nodes)*^{n}, where n is the path loss exponent. As a result, the link weight is proportional to the transmission power for forwarding the data from one node to another. Under this assumption, the α tree algorithm can always find the optimum network topology with the minimum overall energy cost if a fixed compression ratio c is assumed at each forwarding node. The value of α in that case should be set to c, and the proof will be given in the workshop. Figure 2 shows topologies generated with α equals 0 (MST), 1 (MDT), and 0.8 respectively. In this example, 200 sensors are spread randomly over a 30×30 region with a sink at the center, and the path loss exponent is chosen to be four. By visual inspection, the simulation results primarily agree with our expectation. We have also calculated the energy cost of the topologies based on different values of α and n, assuming a fixed compression ratio of 0.8. Table 1 summarizes the results. The energy cost of a topology is defined as follows:

$$\textit{Energy Cost} = \sum_{\textit{all links}} \frac{\textit{No. of bits transmitted on a link after data aggregation}}{\textit{No. of bits in a message}} \times (\textit{Distance})^n$$



(a) (b)



(c)

Figure 2 α tree topology with (a) $\alpha = 0$, (b) $\alpha = 1$, and (c) $\alpha = 0.8$.

Path Loss Exp. (n)	α = 0	α = 1	α = 0.8
2	2302	2763	2118
3	4360	5512	4103
4	8892	10033	8574

Table 1 Overall energy costs of α trees under the assumption of a fixed compression ratio of 0.8

The α tree algorithm makes a pioneer attempt on relating data aggregation performance to the generation of routing topologies which minimize the total energy cost for data funneling. The performance of data aggregation is dependent on the type of application or the nature of the data. The α tree algorithm can easily adapt to the variations in compression performances through the adjustment of a single parameter. In general, the optimum value of α decreases as the amount of data reduction increases.

Work in Progress

In last section, we mentioned that α tree works best for networks having a fixed data compression ratio at each forwarding node. Obviously, such a model is not completely accurate. In the real world, data should have temporal and spatial correlations, e.g. temperature over a region. Therefore, a more realistic data compression model should be developed. We are applying information theory to defining a generic compression model, taking into consideration possible temporal and spatial correlations. Based on the refined compression model, we will then evaluate the performance of α tree, e.g. percentage reduction on total energy cost with respect to node density and sensor-to-sink ratio, as compared to MST and MDT. We will also investigate the relationship between the choice of α and the data aggregation performances. In addition, we may use optimal routing [2] to generate optimal trees and compare these trees with best α trees. Finally, we may study the overhead in generating α trees and find out the response of the α tree algorithm at different levels of node mobility.

References

[1] N.F. Maxemchuk. Video Distribution on Multicast Networks. IEEE JSAC, 15(3): 357-372, April 1997.

[2] D. Bertsekas and R. Gallagher. Data Networks. Prentice-Hall, 1992.