An Optimal Bound for the MST Algorithm to Compute Energy Efficient Broadcast Trees in Wireless Networks

Qassem Abu Ahmad

February 10, 2008

Contents

1 Introduction

A wireless network is a collection of autonomous mobile nodes that communicate with each other over a wireless channel. Such a wireless network without any central administration is called an ad-hoc network, which we will use in the rest of the paper. The nodes in an ad-hoc network are mobile and may cooperate in routing each other's data packets. The transmitted signal from the sender s to the receiver t decays with distance $dist(s,t)$ as $\frac{1}{dist(s,t)^\alpha}$, where α can vary from 1 to more than 6 depending on the environment of the transmission. In vacuum is $\alpha = 2$. The transmitted signal is an electromagnetic wave of power P_s that is emitted in all directions. In order for the receiver t to receive the signal successfully, the arrived signal at t must satisfy the following inequality

$$\frac{P_s}{dist(s,t)^\alpha} > \gamma \tag{1}$$

where γ is the *transmission-quality* parameter, that enables the receiver to reconstruct the arrived signal (we are ignoring here the influence of the ambient noise).

Each node is assigned a power level P that determines according to (1) its **transmission range**, in which other nodes receive the signal successfully. Let $G = (S, A)$ be the **transmission graph**, where S is the set of the nodes or vertices from 1 to n, and the directed edge from s to t is in A if and only if t is within the transmission-range of s. Let ω be the weight function of the graph which is equal to the Euclidean distance between the incident vertices of each edge. Let $r_G(i)$ be the maximal length of the outgoing edges from node i, then every node within $r_G(i)$ can also receive data sent by node i. Let

$$power(G) = \sum_{i \in S} \gamma \cdot r_G(i)^\alpha \tag{2}$$

be the total power needed to establish all connections in G. The value of γ does not influence our results so we will set it to one [1].We describe now the main problem and the goal of this summary.

The Energy Efficient Broadcast Tree Problem(EEBT): Let S be a set of nodes represented by points from the Euclidean plane. That is, the distance function becomes $dist(s,t) := |st|$, where $|st|$ is the Euclidean distance between s and t. One of the nodes is called the source node s. The **goal** is to find the transmission graph G which minimizes $power(G)$ and contains a directed spanning tree rooted at s.

The importance of this problem is because that **broadcasts** are often used in real life, i.e. between base stations of the Telekom in germany. The source node initiates the transmission and sends a message to all other nodes in the graph. The EEBT problem is NP-hard [2, 3], and if the distance function is arbitrary, then the problem can not be approximated with a logarithmic factor. The currently best approximation algorithm for the EEBT problem is as follow:

The MST Algorithm(MSTALG): The input of the algorithm is a set of nodes S represented by points in the Euclidean plane. One of the nodes is designated as the source. The algorithm first computes the Euclidean minimum spanning tree (EMST) of the point set S. Then the MST is turned into a directed EMST by directing all the edges such that there exists a directed path from the source node to all other stations.

This summary shows in section 3 that MSTALG is a 6-approximation algorithm for every $\alpha \geq 2$ and then offers in section 4 a faster distributed protocol for constructing a MST, the **Fast_MST Protocol**.

In the network model in section 4 the communication is synchronous and occurs in discrete pulses, called **rounds**, which will be used as a measure of efficiency. All vertices are synchronized and start working at round 1. The distributed computation works in the **CONGEST(B)** model, in which each vertex is allowed to send a message of maximum size B. We also assume that each station has unlimited capacity of computation.

Each vertex v of the graph should accept the input graph itself (G, ω) and its MST-radius, which we will define later, and should return as output the set of edges that are adjacent to v and belong to the MST of the graph (assuming that the latter is unique; otherwise, the outputs of all the vertices should be consistent, and form an MST). So we would first apply *Protocol Fast_MST* on the input Graph (G, ω) to construct the MST, then the MST is turned into a directed MST by directing all the edges such that there exists a directed path from the source node to all other stations. Then each node would send the message received from s to its neighbors nodes.

In the next section we give the main results of the paper. In section 3 we sketch the proof that the MSTALG is a 6-approximation algorithm of the problem mentioned above. Section 4 offers a faster distributed algorithm forconstructing a MST.

2 Main Results

Theorem 1 *Let S be a set of points from the unit disk around the origin, with the additional property that the origin is in S. Let $e_1, e_2, \cdots, e_{|S|-1}$ be the edges of the Euclidean minimum spanning tree of S. Then*

$$\mu(S) := \sum_{i=1}^{|S|-1} |e_i|^2 \leq 6 \tag{3}$$

Theorem 1 and the next Lemma prove that MSTALG is a 6-approximation algorithm for the EEBT problem.

Lemma 1.1 *A bound on $\mu(S)$ automatically implies the same bound on the approximation ratio of **MSTALG** for $\alpha \geq 2$. [4]*

The next theorem gives an upper bound on the protocol Protocol Fast_MST

Theorem 2 *Protocol Fast_MST computes the MST of an n-vertex graph $(G = (V, E), \omega)$ assuming that every vertex v accepts as input the maximum MST-radius $\mu(G, \omega)$ of the graph in $O(\mu \cdot \log^3 n + \sqrt{n/B} \cdot \log^{3/2} n \cdot \sqrt{\log^* n})$ rounds of* distributed computation with success probability $1 - O(1/poly(n))$.

The next section proves theorem 1. Section 4 proves theorem 2 and introduces how the Fast_MST works.

3 An Optimal bound on the Approximation Ratio of MSTALG

We assume here that *Fast_MST* has already constructed the Euclidean minimum spanning tree of the given transmission graph $G = (S, A)$ with the Euclidean distance as the weight function ω of the graph, that is $\omega(e) = dist(u, v)$, where e $\in A$, and u, v are the incident vertices of e.

The MST is normalized to fit into a unit disk around the origin and one of the vertices in S is the origin of the unit disk. Let $e_1, e_2, , e_{|s|-1}$ be the edges of the constructed Euclidean minimum spanning tree of S. The total power of the normalized MST of G is then as follows:

$$\mu(S) = \sum_{i=1}^{|S|-1} |e_i|^2. \tag{4}$$

We set here $\alpha = 2$, but the results of this section are correct for any $\alpha \geq 2$. We will find now an optimal upper bound on $\mu(S)$, which is according to Lemma 1.1 the same upper bound on the approximation ratio of **MSTALG**.
The proof's idea is similar to the one in [4]. In [4] it was proved that the MSTALG is an 12.15-approximation algorithm. The cost of each edge e was represented by a *diamond*, that is, two isosceles triangles with an angle of $\frac{2\pi}{3}$. All the diamonds around the edges of a MST do not intersect, and the sum of their areas is equal to $\lambda \cdot \mu(S), \lambda = \frac{\sqrt{3}}{6}$ and is bounded by the area of some polygon of area 12.15. Thus, 12.15 is an upper bound on the approximation ratio of MSTALG.

To get a lower upper bound we use a larger geometric shape, but we must then deal with intersection of the shapes. Let the cost of each edge e of the MST be represented by a pair of equilateral triangles of side length $|e|$. The area of this pair is equal to $|e|^2 \cdot \sin 60 = \lambda \cdot |e|^2$, with $\lambda = \frac{\sqrt{3}}{2}$. We need now to find an upper bound of the total area generated by all such pairs of equilateral triangles. Consider the MST of S and the Delaunay triangulation of the convex hull of S. Consider the unique cycle of the convex set that is formed by adding a **door** edge. This cycle with its interior is called a **pocket** P. Each triangle in the Delaunay triangulation is called a **pocket triangle**. All MST edges of the cycle are called **border edges**. The rest are called **interior edges**. Each interior edge e generates two MST-triangles of the pocket P, which are equilateral triangles with side length $|e|$. Each border edge generates a MST-triangle towards the center of the pocket P. The MST-area of the pocket P is the sum of all MST-triangles.
In Figure 1 on the left the dashed line is the only door of the pocket. Each border edge generates one MST-triangle and each interior edge generates 2 MST-triangles. On the right side one can see the Delaunay triangulation of the MST.

We prove now that the number of MST-triangles is equal to the number of pocket-triangles plus one.
Let $n, e, b, t,$ be the number of nodes, edges, border edges and the pocket triangles of a pocket. The following equations hold.

$$n + (t + 1) - e = 2$$
$$2e = 3t + (b + 1)$$

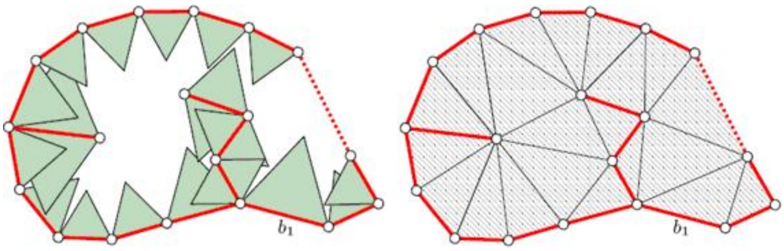

Figure 1: A pocket with its MST-triangles on the left and the pocket triangles on the right. Note that there are 22 MST-triangles and 21 pocket triangles.

The first formula is the Descartes-Euler polyhedral formula. The second formula represents the number of edges of all facets. Solving for t, we get that $t = 2n - 2 - b - 1$. The number of MST-triangles is equal to $2n - 2 - b$ because each interior edge generates two MST-triangles while each border edge generates only one. So we get that the number of MST-triangles is equal to the number of pocket-triangles plus one.

Let the **extended pocket area(EP-area)** of a pocket be the area of the pocket plus an additional equilateral triangle with side length b, where b is the longest edge of the pocket. Then the difference between the EP-area and the MST-area of the pocket is always positive.(see [1] for the proof).
By using the next Lemma we can prove theorem 1.

Lemma 2.1 *Consider a pocket formed by an edge e. Then its MST-area can be bounded by the area of the pocket plus the area of a set of equilateral triangles whose side lengths are bounded by 1 and add up to* $|e|$*.*

Proof of Theorem 1. Consider the pockets whose doors are the edges of the convex hull T of S. The entire MST-area of G is equal to the sum of all MST-areas of all pocktes.

By using Lemma 2.1 we conclude that the entire MST-area is upper bounded by the area of the convex hull plus, for each edge e of T, the area of an equilateral triangle with side length $|e|$. The latter shape is called a **sun**. By moving all the points in T towards the unit circle, the area of its sun will increase and will reach the maximum if all points of T are on the unit circle.

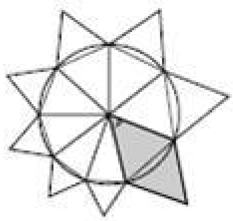

Figure 2: A sun

Figure 2 shows the sun of T with maximum area. The area of each part of the sun (the dashed part) can be expressed in terms of the angle ρ.

$$f(\rho) = 0.5sin(\rho) + \sqrt(3) \cdot sin^2(\rho/2).$$

Since we have assumed in Lemma 2.1 that each edge of the convex hull is bounded by one, the angle ρ must be between 0 and $\pi/3$. It follows that the maximum area of the dashed part is achieved when $\rho = \pi/3$. the MST-area of the pockets $= \sin 60 \cdot \mu(S) = \sin 60 \cdot \sum_{i=1}^{|S|-1} |e_i|^2 \leq 6 \cdot f(60) = 6 \cdot sin60 \rightarrow \mu(S) \leq 6$.

4 A Faster Distributed Protocol for constructing a MST

The *Protocol Fast_MST* consists of two parts. First it constructs a MST-forest, then it execute a sub-procedure in parallel through each tree τ of an appropriate collection of trees that satisfy certain properties. This sub-procedure removes unnecessary edges in τ that do not belong to the MST-forest. We need to define some parameters, that we will need later on. The distance between a vertex u and an edge $e = (v, w)$, denoted $dist_G(u, e)$, is defined as the minimum of the distances between u and v, or between u and w. The *vertex-edge radius* of the cycle C w.r.t. the vertex u, denoted $VxEdgRad_u(C)$. Let $Z(e)$ denote the set of the cycles C in which the edge e is the heaviest edge. Let $e = (u, v)$, we define $VxEdgElimRad(u, e) = min\{VxEdgRad_u(C)|C \in Z(e)\}$ (The minimum of an emapy set is 0). The *MST-rdius* of (G, w) w.r.t. the vertex u, denoted $\mu_u(G, \omega)$, is defined by $\mu_u(G, \omega) = max\{VxEdgElimRad(u, e)|e \in E, u \in e\}$. The *MST-radius* of (G, ω) is the maximum of the set $\{\mu_1(G, \omega), \cdots, \mu_n(G, \omega)\}$, where $V = 1 \cdots n$

A k-MST forest F of a Graph $(G = (V, E), \omega)$ is a collection of vertex disjoint trees of depth $O(k)$ and the number of vertices in each tree T is $\Omega(k)$. Each Tree $T \in F$ is a fragment of the MST of (G, ω), that is, a connected subtree of G with minimal cost w.r.t. the restriction of G to $V(T)$. A k-MST forest can be constructed in $O(k \log^* n)$ rounds of distributed computation, where τ is an auxilary tree that contains each $e \in \hat{E}$.

We define the multigraph $\hat{G} = (\hat{V}, \hat{E})$ with $\hat{V}$ is the set of the fragments of the constructed k-MST forest, and $\hat{E}$ is the set of the inter-fragment edges. The acyclic subset of the edge set of $\hat{G}$ forms a matroid of rank $|\hat{V}| - 1$, where $\hat{E}$ is the universe of the matroid. The distributed *Protocol Pipeline* [5] computes a maximum independent set with minimum weight of a matroid with universe $\hat{E}$ under the *CONGEST(B)* model in running time of $O(depth(\tau) + |\hat{V}|/B \cdot \log n)$ rounds. The aim now is to construct a MST of $\hat{G}$ by computing a maximum independent set of the matroid of minimal weight. This is done by applying the *Protocol Pipeline* on a collection C of auxilary trees that satisfy the following properties:

1. Each tree τ of $depth(O(W \cdot \kappa))$, with $W = \mu(G, \omega)$ and $\kappa = \log n$.

2. Each $v \in V$ appears in at most $O(\kappa \cdot polylog(\kappa))$ different trees of C.

3. For Each $v \in V$ there exists a tree in C that contains the entire W-neighborhood of v.

We call such a collection the (κ, W)-*neighborhood cover* or shortly (κ, W)-*cover of* G. Let the *cover-degree* of a vertex v be the number of clusters of C that contains the vertex v. In [5] a ***Protocol Cover*** was devised to construct a (κ, W)-*cover* with maximum *cover-degree* $O(\kappa \cdot n^{\frac{1}{\kappa}} \log n)$ with running time $O(\kappa^2 \cdot \log n \cdot n^{\frac{1}{\kappa}} \cdot W)$ with probability $1 - O(1/poly(n))$.

Recall that the algorithm first constructs a *k-MST* forest F, then it constructs $(\log n, \mu)$-*cover* C. The *Protocol Pipeline* is applied in each tree τ in C so that at the end we get the MST of G.

For each $\tau \in C$ let G^τ and F^τ be a restriction of G and F to the vertex set of τ respectively. Observe that the fragments $T_1, \cdots, T_p$ of F will not be necessarily connected after the restriction of F to the vertex set of τ. Let ${T_i}^\tau$ denote the resulting forest by restriction $V(T_i)$ to $V(\tau)$ with the edge set $\{e = (u, w) \in E(T) | u, w \in V(\tau)\}$. Consider the multigraph $\hat{G}^\tau$ with T^τ serving as vertex for each fragment T^τ. For each inter-fragment edge (u, v) with $u \in V({T_1}^\tau)$, $w \in V({T_2}^\tau)$, there is an edge between ${T_1}^\tau$ and ${T_2}^\tau$ in $\hat{G}^\tau$.
Next, *Protocol Pipeline* is invoked on each tree τ in C to compute the MST of $\hat{G}^\tau$. Let *Protocol Pipe_MST* denote the computation till this stage. Observe that the number of supervertices of $\hat{G}^\tau$ is at most $O(n/k)$, because each fragment of F yields at most one not necessarily connected supervertex of $\hat{G}^\tau$. Hence, the number of the MST edges of $\hat{G}^\tau$ is also $O(n/k)$. Therefore, *Protocol Pipeline* computes the MST of $\hat{G}^\tau$ in $O(depth(\tau) + \frac{n \cdot \log n}{k \cdot B}) = O(\mu \cdot \log n + \frac{n \cdot \log n}{k \cdot B})$ rounds if no node terminates while the computation.

Since each vertex v may participate in $O(\log^2 n)$ trees in C, $O(\log^2 n)$ trees will try to access the vertex v. Hence, the running time of *Pipeline_MST* will increase at most by factor $O(\log^2 n)$, that is, $O(\mu \cdot \log^3 n + \frac{n \cdot \log^3 n}{k \cdot B})$.
We are constructing now by a new protocol called *Cover_MST* the MST of the original Graph out of the *k-MST* forest F, and the MSTs of the multigraphs $\hat{G}^\tau$ for all trees in C.
Each vertex v decides locally for each incident edge e, if e belongs to the MST of G. The edge e belongs to the MST of G if e belongs to the $k - MST$ forest F or e is an inter-fragment edge and also belongs to the MST of each multigraph $\hat{G}^\tau$. In other words, let C be a cover that coarsens the (κ, μ)-*cover*, then there exists a tree τ in C with $u, w \in V(\tau)$, so that e does not belong to the MST of $\hat{G}^\tau$.
We sketch now the proof of the last statement.

Consider an edge $e = (u, w)$ that does not belong to the MST of G. By definition $\mu \geq VxEdgElimRad(u, e) = VxEdgRad_u(C_0)$ for some cycle C_0, in which the edge e is the heaviest edge. By definition of the (κ, W)-*cover* (Property 3) there exists a tree t in the (κ, W)-*cover* that contains the whole μ-*neighborhood* of each vertex, here vertex u. it follows that the whole μ-*neighborhood* of u is included in some tree τ in C (that follows from the definition of a coarsen). Recall that e is an inter-fragment edge. This implies that there exists a cycle C_0^τ in the multigraph $\hat{G}^\tau$, in which e is the heaviest edge. It follows that the edge e does not belong to the MST of $\hat{G}^\tau$.
Thus, Theorem 2 is proved.

References

[1] Christoph Ambuehl. An Optimal Bound for the MST Algorithm to Compute Energy Efficient Broadcast Trees in Wireless Networks, pages 1139-1150. Department of Computer Science, The University of Liverpool.

[2] A.E.F. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca. On the Complexity of Computing Minimum Energy Consumption Broadcast Subgraphs. In *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 121-131, 2001.

[3] A.E.F. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca. A Worst-case Analysis of an MST-based Heuristic to Construct Energy-Efficient Broadcast Trees in Wireless Networks. Technical Report 010, University of Rome Tor Vergata, Math Department, 2001.

[4] P.J. Wan, G. Calinescu, X.Y. Li, and O. Frieder. Minimum-Energy Broadcast Routing in Static Ad Hoc Wireless Networks. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1162-1171, 2001.

[5] Michael Elkin, A Faster Distributed Protocol for Constructing a Minimum Spanning Tree, page 359-368. *In Proc. of the 15th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA), New Orleans, Louisiana, USA, January 2004.*