A self-stabilized distributed algorithm for the range assignment in ad-hoc wireless networks*

G. Huiban[†]
Departemento de Ciência da Computação
Universidade Federal de Minas Gerais
Avenida Antônio Carlos, 6627
31270-901 Belo Horizonte - MG - Brasil
ghuiban@dcc.ufmg.br

Y. C. Verhoeven
Dipartimento di Matematica
Università degli Studi di Roma "Tor Vergata"
Via della Ricerca Scientifica, 1
00133 Roma, Italy
verhoeve@mat.uniroma2.it

Abstract

In this paper we consider the problem of computing a range assignment in an ad-hoc wireless network which allows a specified source station to perform an energy efficient broadcast operation. This problem has been showed to be NP-complete and several approximation schemes have been proposed. In particular, one make use of the minimum spanning tree of the network. We show here that this approximation can be performed using a self-stabilizing distributed algorithm using a polynomial number of messages. Such an algorithm is particularly interesting in this context as a wireless network seems naturally distributed.

keywords : ad-hoc wireless network, range assignment, distributed algorithm, self-stabilization

1 Introduction and problem

At the present time, wireless networks face a tremendous growth in interest largely due to the spectacular drop in hardware prices: cellular phones are now widely spread, and companies do not hesitate to invest for having a wireless intra-net. There exists two kinds of wireless networks, the *single-hop* architecture which uses a static backbone while terminal links are wireless [15] (*e.g.:* the cellular phones networks), and the *multi-hop* networks which do not rely on any fixed infrastructure nor wired part [16]. *Ad-hoc* networks may be the most popular type of the latter architecture, all the stations have the same role and importance. In the rest of the paper we will only deal with multi-hop ad-hoc networks.

It can be easily viewed that ad-hoc wireless networks are inherently distributed. This aspect, which is a key feature, makes them very flexible, adaptive and allow their use anywhere without further needs of infrastructure. One could think of the various advantages of such a network architecture in various cases, such as organization of assistance after a natural disaster. Indeed, this makes centralized algorithms rather unnatural in multi-hop networks but to our knowledge very few distributed algorithms have been proposed so far (see for example [5]).

Unfortunately, those networks are subject to the problem of *interferences* (or *collisions*), which happen when a station receives at the same time several messages from several different stations [12]. This problem is emphasized with the lack of centralized structure and has to be taken into account when developing algorithms for ad-hoc wireless networks. It must be remarked that recently developed protocols take care of this problem by assigning different frequencies to the different stations. In our opinion, this strategy limit the scaling of the networks as there must be a minimum distance between two used frequencies.

Each station has respectively a *maximum transmission range* and an *effective transmission range* which correspond to the maximum and effective distance possible to reach directly another station: the maximum range is defined by the physical characteristics of the station whereas the effective range

*This work was supported by the European project ARACNE
[†]contact author.

(necessarily lower or equal than the maximum range) corresponds to a maximum distance reached by messages in regards with the setting of power devoted to transmission. For a set of stations S, we call *range assignment* the function $r : S \to \mathbb{R}$, which for each station s define the transmission range of s. The maximum (*resp.* effective) transmission range of s will be noted $R_{max}(s)$ (*resp.* $R(s)$).

Also remark that the effective range may not be sufficient for covering all stations in the network. This is not a problem: it suffices to send a message to another station, which will relay the message and send it to another station, and so on until the message reaches its final destination.

1.1 The Range assignment problem

As wireless devices are often small and portables, energy consumption is one of the key problem in ad-hoc wireless networks. It must be remarked that the power consumed by the station depends on the transmission range. In particular in the most common models, the power $P(s)$ supplied by a station s in order to reach the station t must satisfy the inequality

$$\frac{P(s)}{d^{\alpha}(s,t)} > \gamma,$$

where α (the *distance-power gradient*) and γ (the *transmission quality*) are constants. In an ideal environment, it holds that $\alpha = 2$, but this value can vary from 1 to 6, depending on the environment conditions (see [14]). Thus, the effective range of a station depends of its power and the energy cost needed for a transmission is proportional to $cost(u) = (R(u))^{\alpha}$. Indeed, the total cost for the network is proportional to,

$$cost = \sum_{\text{every station } s} (R(s)^{\alpha}).$$

Thus one might be willing to reduce the transmission range in order reduce the energy consumption, while keeping the connectivity properties of the network so that the stations can still communicate. Choosing the best (in term of cost) transmission range among all possible ones so that the network respects a certain property π is a combinatorial problem, known as the *range assignment problem for the property* π.

1.2 Using Minimum Spanning Tree based algorithms

A wireless network is often modelised by a weighted directed graph embedded in a Euclidean space (this graph is called the *communication graph* and is noted G_r). In the context established above, with $\alpha > 1$ and the dimension *dim* of the Euclidean space greater than 2, the range assignment problem has been proven to be NP-complete [11, 9].

Different approximations schemes have been proposed. One in particular uses the *Minimum Spanning Tree* (in short mst) of the communication graph to set the range for each station. From a minimum spanning tree, it is very simple to get an approximation of a solution for the range assignment problem: for each node, the effective range is set equal to the longest outgoing edge [7].

Fast (polynomial-time) algorithms solving the minimum spanning tree problem exist, such as Kruskal's algorithm or Prim's algorithm [2]. Setting ranges from a minimum spanning tree is also polynomial. Hence, this algorithms runs in polynomial time, and as simple as it appears, it provides a rather interesting solution. Depending on the property wanted to be achieved for the graph, some results have been proven. For a strongly connected graph, the solution found is at most 2 times the value of the optimal [7]. In the case of broadcasting capabilities, it has been proven that such an algorithm provides at most a ratio of 12 [17].

1.3 Distributed algorithms and wireless networks

As seen previously, ad-hoc wireless networks contain no fixed infrastructure nor centralized coordinator; each station have to be able to organize its relationship with the whole network without relying on a central authority. From this point of view, distributed algorithms are particularly adapted to ad-hoc wireless networks. Several distributed algorithms for computing a minimum spanning tree have been proposed (see [1, 4]).

Meanwhile the development of a distributed algorithm for the broacast in wireless network has been presented as a research challenge ([3, 18]), only one heuristic have been proposed (see [6]). In this paper we present a fully distributed algorithm for the range assignment problem. This algorithm rely on the distributed minimum spanning tree algorithm described in [4]. The main difference between the two algorithms lies in the way the communication are done.

It is a crucial point to be able to adjust to a change in the topology of the network. The algorithm presented in this paper is a self-stabilized algorithm; which means that each node constantly runs a rule and adjust itself to its environment. In the case of node failure, the solution will adjust to the change of topology.

One can also remark that the ability to adjust the minimum spanning tree to the changes of the network enables us to consider also sets of mobile hosts, provided that the movements are slow compared to the speed of the stations.

Also note that in [4], there is neither time complexity analysis of the algorithm nor any analysis of the number of messages exchanged. Nevertheless, the algorithm is proven to complete in finite time. We provide a worst case analysis for the range assignment algorithm provided here in the particular setting of a wireless network, and believe that this analysis can be extended to less-specific context for the algorithm in [4].

We chose to concentrate on the particular synchronous time model. It is known that this model is not realistic, but it enables us to lower bound the execution time of the algorithm, whatever the chosen time model is. It is to be remarked that in ad-hoc wireless networks, this particular time model has been shown to be almost optimal for the broadcast operation (see [8]).

In this paper we prove the following theorem:

Theorem 1 *There exists a distributed algorithm such that in a wireless ad-hoc network with n stations and without edge-fault, each station knows which of its edges belong to the minimum spanning tree in time $O(n(n-1)^2)$.*

1.4 Notations

Let S be a set of stations, $|S| = n$. Each station represent a node of G_r and there exists an edge from a node x to another node y if and only if the station represented by x can directly (in only one hop) send a message to the station represented by y. The weight of an edge (x, y) is defined as $d(x, y)^2$ where $d(,)$ is the Euclidean distance.

Let u be a node of V and consider all the vertices at distance less than $R_{max}(u)$ (*resp.* $R(u)$). Let's $N_{max}(u)$ (*resp.* $N(u)$) be the set $N_{max}(u) = \{v \in V, v \neq u/d(u,v) \leq R_{max}()\}$ (*resp.* $N(u) = \{v \in V, v \neq u/d(u,v) \leq R(u)\}$).

Let $E_{max} = \{(u,v)/u \in V, v \in N_{max}(u)\}$ (*resp.* $E = \{(u,v)/u \in V, v \in N(u)\}$) be a set of edges. We will alternatively use the notations $(u,v) \in E_{max}$ and $e \in E_{max}$ for edges and for derivate notations ($w_{u,v}$ and w_e for the associated weight). $G = (V, E)$ represents the effective communication graph.

Each edge $(u,v) \in E$ is supposed to have a unique associated weight $w_{u,v} = d(u,v)^\alpha$. Note that this is not restrictive: it is always possible to add lexicographic information to make them unique. The unicity of edge weight ensure the unicity of minimum spanning tree (see [10]). Each station "knows" the topology of the network up to its maximum transmission range, and have an upper bound of the number of node in the whole network.

Consequently, $G_{max} = (V, E)$ is a directed symmetric graph in an Euclidean space, with no self loop or parallel edges. We will also consider G_{max} as connected. If not, this means that the maximum range allowed for the nodes is not sufficient to produce a connected graph.

For any couple of vertices $(u,v) \in V^2, v \neq w$, there exists a elementary path P going from node v to node w. We define α-cost$(P) = max_{e \in P}\{w_e\}$, and $\Psi_{u,v} = \min_P\{\alpha\text{-cost}(P)\}$. α-cost(P) stands for the maximum of the weights of edges belonging to a path P, and $\Psi_{u,v}$ for the minimum of such α-cost for all paths.

With those notations, the problem naturally becomes the aim of minimizing $\min \sum_{u \in V} cost(u)$ such that G satisfies to some particular properties.

2 Algorithm

As previously described, this range assignment algorithm is mainly based on the distributed minimum spanning tree algorithm described in [4]. But in order to be applied with ad-hoc wireless networks, some modifications are needed, in particular in that environment, one has to deal with the communication in order to avoid collisions of messages. We address this topic in the following.

2.1 Hypothesis

The following theorem was proved in [13] and establish a relation between the minima of the α-costs of the paths and the minimum spanning tree of the graph.

Theorem 2 *Let $G = (V, E)$ a connected undirected symmetric graph, with unique edge weight. An edge $\{u, v\}$ is in the unique minimum spanning tree if and only if $\Psi_{u,v} = w_{u,v}$.*

Thus, it is only required to compute all the α-costs in order to determine the minimum spanning tree. As we are now showing, the α-costs can be computed locally by every node in a distributed way.

The algorithm works as follow: each node u tries to compute the value of $\Psi_{u,v}$ for all his neighbors v and the edges belonging to minimum spanning tree are those verifying theorem 2. Note that this implies that each node know the number of nodes in G. Note also that with the algorithm that we propose, each node only have a local knowledge of the network.

Let be M_D a positive numbers such as $\forall e \in E, M_D > w_e$, and for any pair $(u, v) \in V^2$ we define:

- $D_u(v)$ the current estimation of $\Psi_{u,v}$ at node u.
- $S_u = \{i \in V / D_u(i) < M_D\}$. In the case of $S_u \neq \varnothing$, we can define:
  - $\forall i \in S_u, \delta_{u,i,v} = \max(w_{u,i}, D_i(v))$
  - $\delta_{u,v}^{min} = \min_{i \in S_u} \{\delta_{u,i,v}\}$

Each node locally keep a copy of matrix D. By *turn* we mean the right to send message: to avoid collision, nodes must not send data at anytime. Here we make use of the *round-robin* strategy, that is, only one node is allowed to send a message at a particular time and every node waits for its turn to send a message. Nevertheless, all the nodes are able to receive a message at any time. In our algorithm a node will know when it is its turn to send a message thanks to the boolean variable named *turn*. This *turn* may be "clock based", that is every t seconds the node is allowed to send message (which is completely distributed as far as the devices are synchronized).

Note that latest wireless transmission protocols may avoid this problem of collision.

2.2 Rule

Each node $u \in V$ applies this algorithm continuously.

```
if turn then
  for v ∈ V, v ≠ u do
    Send (D_u(v))
  end for
end if
for v ∈ V do
  if u = v then
    D_u(v) = 0
  else
    Compute S_u
    if S_u = ∅ then
      D_u(v) ← M_D
    else
      for w ∈ S_u do
        Compute δ_{u,w,v} ← max(w_{u,w}, D_w(v))
      end for
      Compute δ^{min}_{u,v} ← min_{w∈S_u} {δ_{u,w,v}}
      D_u(v) ← δ^{min}_{u,v}
      Compute Ω_u ← {w ∈ N(u)/w_{u,w} = D_u(w)}
      R(u) = max_{w∈Ω_u} {d(u,w)}
    end if
  end if
end for
```

This algorithm is proved to finish in finite time in [4], that is every node will reach a stable state after a certain time. Our use of the *turn* enables us to give an upper bound of this time. Moreover, one can remark that such an algorithm is particularly suited for mobile hosts, as the nodes move and their neighborhood changes, the algorithm updates the different values and keep the local knowledge of the minimum spanning tree up to date.

3 Analysis of the algorithm

3.1 Complexity analysis

For the complexity analysis, we consider that the iteration of *turn* takes place regularly, allowing each station to run the algorithm fairly.

In the following, we will write $D_u(v)(k)$ the value of $D_u(v)$ during the kth iteration of the main loop of the algorithm.

Lemma 1 *Let $G = (V, E)$ be a graph and* $\mathsf{mst}(G)$ *the associated* MINIMUM SPANNING TREE. *For any (u, v) in V^2, let P be the unique path from u to v in* $\mathsf{mst}(G)$. *If l is the number of edges of P, then $D_u(v)(l) = \Psi_{u,v}$.*

PROOF. A proof of theorem 1 can be achieved by recurrence on l. For any node, we can consider that the values of the parameters are as follows,

$$\forall (u, v) \in V^2 : \begin{cases} D_u(v)(0) = 0 & \text{if} \quad u = v \\ D_u(v)(0) \in \{\Psi_{u,v}, M_D\} & \text{if} \quad u \neq v \end{cases}$$

Let u and v be two nodes of G. Let $\mathsf{mst}(G)$ be the minimum spanning tree associated with G.

Case $l = 1$: in that case the path from u to v in $\mathsf{mst}(G)$ has only 1 edge (note that $v \in N(u)$ and $v \in S_u$). By definition, $D_u(v)(1) = \min_{i \in S_u} \{\delta_{u,i,v}(0)\}$. But for any $i \in S_u$, $i \neq v$ we have $\delta_{u,i,v} = M_D$ and $\delta_{u,v,v} = w_{uv}$ (as said above $D_v(v)(0) = 0$). Thus we have that $D_u(v) = w_{uv}$ that is to say $D_u(v)(1) = \Psi_{uv}$ according to theorem 2.

Case l: Here the path between u and v along the minimum spanning tree has l edges. Let w be the node such that the path from w to v in $\mathsf{mst}(G)$ has $l - 1$ edges and the edge (u, w) belongs to $\mathsf{mst}(G)$.

We have $D_u(v)(l) = \min_{i \in S_u} \{\delta_{u,i,v}(l-1)\}$ and $\delta_{u,w,v}(l) = \max(w_{u,w}, D_{w,v}(l-1))$. The recurrence hypothesis gives us $D_{wv}(l-1) = \Psi_{w,v}$. Then $\max(w_{uw}, \Psi_{w,v}) = \Psi_{u,v}$ which implies $D_{uv}(l) = \Psi_{u,v}$. □

In the previous lemma we have determined how long it takes for each node to get the necessary information to construct the minimum spanning tree. We now have to show that once this information is known, it won't change, provided the graph stays still. This behavior is ensured by lemma 24 of [4].

Lemma 1 implies that information propagates along to the minimum spanning tree. Thus, for any pair of nodes $\{u, v\}$ after a number of iterations equal to the length of the path between those two nodes in the minimum spanning tree each one will have a correct estimation of the α-cost(). In the meantime, each node w of this path will acquire a correct estimation of the values of the α-cost( ) of the path between w and u and w and v.

If h is the diameter (in number of edges) of the minimum spanning tree; after h iteration of the algorithm, each node will have a correct view of the minimum spanning tree. If we consider that each message is sent during a time-slot. During one iteration, each node sends $n - 1$ messages. As the *turn* has to cover each node; that is one iteration takes $O(n(n-1))$ time-slot. And the whole time complexity is $O(hn(n-1))$. Note that in the worst case, it is $O(n(n-1)^2)$.

Theorem 1 is then proved. It is to be remarked that with this scheme, the number of exchanged messages is also of order $O(n(n-1)^2$.

4 Conclusion

In this paper we have modified an existing distributed algorithm to enable it to be used in ad-hoc wireless networks. Moreover we have given an estimation of the time needed by this algorithm to finish. Last we have given some arguments showing that this algorithm is fault-tolerant and thus of considerable use in a network of mobile stations.

References

[1] S Aggarwal and S Kutten. Time optimal self-stabilizing spanning tree algorithm. In *FSTTCS93 Proceedings of the 13th Conference on Foundations of Software Technology and Theoretical Computer Science, Springer-Verlag LNCS:761*, pages 400–410, 1993.

[2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall PTR, february 1993.

[3] J. E. Wieselthier andG. D. Nguyen and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *IEEE INFOCOM 2000*, pages 586–594, Tel Aviv, Israel, 2000.

[4] G. Antonoiu and P.K. Srimani. Distributed self-stabilizing algorithm for minimum spanning tree construction. Technical report, Colorado State University, may 1997.

[5] B. R. Badrinath, Arup Acharya, and Tomasz Imielinski. Structuring distributed algorithms for mobile hosts. In *International Conference on Distributed Computing Systems*, pages 21–28, 1994.

[6] Mario Cagalj, Jean-Pierre Hubaux, and Christian Enz. Minimum-energy broadcast in all-wireless networks: Np-completeness and distribution issues. In *The Eighth ACM International Conference on Mobile Computing and Networking (MobiCom 2002)*, 2002.

[7] A. Clementi, A. Ferreira, P. Penna, S. Perennes, and R. Silvestri. The minimum range assignment problem on linear radio networks. In LNCS 1879, editor, *SA'00; 8th Annual European Symposium on Algorithms*, pages 143–154, 2000.

[8] A. Clementi, A. Monti, , and R. Silvestri. Round robin is optimal for fault-tolerant broadcasting on wireless networks. In *9th European Symposium on Algorithms (ESA'01)*, September 2001.

[9] A. Clementi, P. Penna, and R. Silvestri. On the power range assignment problem on radio networks. Technical Report TR00-054, ECCC, 2000.

[10] E. Horowitz and S. Sahni. *Fundamentals of Computer Algorithms*. Computer Science Press, 1984.

[11] L. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. In *14th Symposium on Theoretical Aspects of Computer Science*, pages 363–374, 1997.

[12] B. Leiner, D. Nielson, and F. Tobagi. Issues in packet radio network design. In *Proceedings of the IEEE*, volume 75, pages 6 – 20, January 1987.

[13] B. M. Maggs and S. A. Plotkin. Minimum-cost spanning tree as a path-finding problem. *Information Processing Letters*, 26(6):291–293, jan 1988.

[14] K. Pahlavan and A.H. Levesque. *Wireless Information Networks*. Wiley-Intersince, 1995.

[15] D. Raychaudhuri and N. Wilson. Atm based transport architecture for multiservices wireless personal communication networks. *IEEE Journal of Selected Areas of Communications*, 12(8), october 1994.

[16] M. Steenstrup, editor. *Routing in communication networks*, chapter Packet radio routing, pages 351–396. Prentice-Hall, 1995.

[17] P. Wan, G. Calinescu, X. Li, and O. Frieder. Minimum-energy broadcast routing in static ad hoc wireless networks. In *IEEE INFOCOM*, 2001.

[18] P.-J. Wan, G. Călinescu, X.-Y. Li, and O. Frieder. Minimum energy broadcast routing in static ad hoc wireless networks. In *IEEE INFOCOM'2001*, Anchorage, Alska, April 23-28 2001.