Energy-Efficient Broadcasting in Ad-Hoc Networks: Combining MSTs with Shortest-Path Trees *

Paolo Penna
penna@dia.unisa.it

Carmine Ventre
ventre@dia.unisa.it

Dipartimento di Informatica ed Applicazioni "Renato M. Capocelli"
Università di Salerno
via S. Allende 2, I-84081 Baronissi (SA), Italy

ABSTRACT

We investigate the problem of constructing a multicast tree in *ad-hoc* networks. In particular, we address the issue of the *power consumption*, that is, the overall energy that the stations must spend to implement such a tree. We focus on two extreme cases of multicast: *broadcast* (one-to-all) and *unicast* (one-to-one). Minimum Spanning Trees (MSTs) and Shortest-Path Trees (SPTs) yield optimal solutions for broadcast and unicast, respectively. Unfortunately, they do not guarantee any optimality for the "counterpart", that is, MSTs are non-optimal for unicast, while SPTs are non-optimal for broadcast.

In this work, we experimentally evaluate the performances of an algorithm combining MST solutions with SPT ones. Our approach is based on the construction of Light Approximate Shortest-path Trees (LASTs) of a given directed weighted graph, introduced by Khuller *et al* [1995]. LASTs approximate *simultaneously* the cost of the MST and the distances of the SPT rooted at a source node, thus yielding, also in the worst case, optimal solutions for both unicast and broadcast.

Rather surprisingly, this "compromise" between MSTs and SPTs, has a very good performance w.r.t the broadcast tree obtained from a MST. Indeed, for randomly-generated instances, the broadcast tree obtained with LASTs is in some cases better (and never much worse) than the broadcast tree obtained from MSTs. This important fact shows that LASTs are not only interesting in theory, but they have practical relevant applications. Indeed, their use in our experiments also provides new insights on the approximation ratio of the MST broadcast algorithm for randomly-generated instances.

*Work supported by the European Project IST-2001-33135, Critical Resource Sharing for Cooperation in Complex Systems (CRESCCO).

Categories and Subject Descriptors

F.2.2 [**Theory of Computation**]: Nonnumerical Algorithms and Problems—*graph algorithms, network problems*

General Terms

Algorithms, Experimentation, Theory

Keywords

Wireless Ad Hoc Networks, Energy Consumption, Multicast Trees, Light Approximate Shortest-path Trees.

1. INTRODUCTION

One of the main benefits of ad-hoc wireless networks relies in the possibility of communicating without any fixed infrastructure. Indeed, each station is a radio transmitter/receiver and communication between two stations that are not within their respective transmission ranges can be achieved by *multi-hop* transmissions: a set of intermediate stations act as relays and forward the message till its destination. Due to the limited power of the stations, multi-hop transmissions are, in general, unavoidable. Moreover, they often result in a significant reduction of the overall energy required by the communication. This is accomplished by suitably varying the transmission ranges of the stations depending on the environmental conditions and on the relative positions of the stations.

In particular, *omnidirectional* antennas are used by all stations to transmit and to receive the messages. These antennas are attractive in their broadcast nature. Indeed, a single transmission by a station can be received by many stations that are "close enough". Consider the example in Fig. 1. The source station s wants to broadcast the a message to all other stations. Because of signal attenuation, the *power* required by s to directly transmit a message to another station j is proportional to $d(s,j)^\alpha$, where $\alpha > 1$ and $d(s,j)$ is the distance between the two stations (see Sect. 1.1 for a more in-detail description of the model). Therefore, the cost of one-hop transmissions is superlinear in the distance between the stations. Also notice that, the power of a station determines its *transmission range*: if station s transmits with power d^α, then all stations at distance at most d from s receives the message (see Fig. 1). Notice that, because of signal attenuation, in general, a single hop transmission from s to *all* other stations may result in a very high power consumption. It is therefore preferable to use intermediate

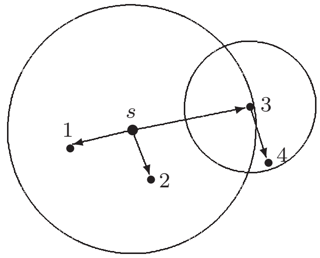


Figure 1: An example of wireless network in the two-dimensional Euclidean space. Disks represent the regions where the attenuated power of the corresponding signal is not smaller than γ: station s transmits with power $d(s,3)^\alpha$ and reaches stations 1, 2 and 3, while transmission from s to 4 is performed in two hops via station 3.

stations that, together with the source s, implement a suitable multicast tree. The overall energy required to perform the transmission is proportional to the sum of the costs of the transmission ranges (i.e., the sum of the energy spent by every station to transmit). For instance, the multicast tree in Fig. 1 requires energy $d(s,3)^\alpha + d(3,4)^\alpha$, that is, the energy consumption corresponding to the two transmission ranges.

We consider the problem of constructing multicast trees requiring *minimal power consumption*[1] (see Sect. 1.1 for a formal definition of the problem). As first observed in [12], energy-efficient solutions must consider the power/range assignment (i.e., a network connectivity, associated to the Physical layer) and the actual construction of a multicast tree (i.e., a routing function, associated to the Network layer) *jointly*. We address the issue of designing centralized (as opposed to distributed) algorithms for constructing energy-efficient multicast trees. This is considered a challenging algorithmic problem even when assuming that the station positions do not vary over time (i.e., static ad-hoc networks). Indeed, computing a minimum-cost solution is NP-hard, thus implying that no polynomial-time (centralized/distributed) algorithm can guarantee $r = 1$, unless P=NP [13] (see the discussion in Sect. 1.2).

Centralized algorithms for the static case are the basis for the design of (more sophisticated) distributed and/or dynamic algorithms. One desideratum is to have some guarantee that the cost of the computed solution is "sufficiently close" to the optimum. Towards this end, two important factors should be considered when designing such algorithms: (i) the *worst case* approximation guarantee, and (ii) the "experimental" approximation performances when restricting to real instances. Clearly, a good algorithm should have a good worst case approximation guarantee and an *even better* approximation behavior when restricting to real instances.[2]

In this work, we present a new algorithm based on a "compromise" between the minimum-spanning tree and shortest-path tree (see Sect. 1.3 for the details). Shortest-path trees and minimum spanning trees provide optimal solutions for unicast and broadcast, respectively (see the discussion in Sect. 1.2). However, minimum spanning trees (resp., shortest-path trees) do not guarantee optimal solutions for unicast (resp., broadcast). Our solution is based on the idea of approximating simultaneously those two problems using so called *Light Approximate Shortest-path Trees* (LASTs) introduced by [14]. This guarantees, in the *worst case*, a constant approximation for *both* unicast and broadcast. Moreover, in some cases, it *improves* over the broadcast tree obtained with minimum spanning trees. We indeed evaluate our algorithm on randomly-generated instances and observe that, surprisingly, the broadcast tree obtained with LASTs is often not worse than the one obtained with a minimum spanning tree. This is somewhat counterintuitive, since our algorithm increases the weight of a minimum spanning tree (which is good for broadcasting) to attain a good performance also w.r.t. the single source-destination distances (unicast).

Our work deals with the problem of computing a solution once the geographical position of the stations is given and we do not consider the effect of stations mobility on the computed solution (i.e., the case of static ad-hoc networks).

1.1 Model and Assumptions

We consider a set $\mathcal{S}$ of stations that are located on the two-dimensional Euclidean space and we represent each station as a point corresponding to its position. A station i is able to *directly* transmit to a station j if and only if the power P_i used by station i to transmit satisfies

$$\frac{P_i}{d(i,j)^\alpha} \geq \gamma,$$

where $d(i,j)$ is the Euclidean distance between i and j, $\alpha \geq 1$ is the *attenuation parameter* depending on the environmental conditions [16] (e.g., in the empty space $\alpha = 2$), and γ is the *transmission quality parameter*: station j will receive the message with attenuated power equal to $P' = P_i/d(i,j)^\alpha$, and the message can be corrected interpreted if $P' \geq \gamma$ (typically γ is normalized to be 1). The power P_i determines the *transmission range* of station i as follows. For every station j such that $d(i,j) \leq \sqrt[\alpha]{P_i/\gamma}$, it holds that $P_i/d(i,j)^\alpha \geq \gamma$, thus implying that j receives the signal sent by i. So, station i is directly connected to all stations located inside the disk centered at i and with range $r(i) := \sqrt[\alpha]{P_i/\gamma}$ (see the example in Fig. 1). Because of this, $r(i)$ is denoted to as the transmission range of station i. Observe that, we are not considering obstacles between stations. So, with the cost (i.e., power) of a single transmission, a station can actually transmit to *all* stations within its transmission range.

Given a set of stations $\mathcal{S}$ and a *range assignment* for it, i.e., a function $r : \mathcal{S} \to \mathbf{R}^+$, the *overall power consumption* is equal to

$$\mathsf{COST}(r) := \sum_{i \in \mathcal{S}} r(i)^\alpha. \tag{1}$$

A range assignment naturally implements a *communication graph* defined as the directed graph $G_r = (\mathcal{S}, \mathcal{E}_r)$, where $\mathcal{E}_r := \{(i,j) | \ d(i,j) \leq r(i)\}$. Interestingly, given a communication graph $C = (\mathcal{S}, \mathcal{E}_C)$, one can obtain a range assignment $r_C(\cdot)$ implementing C by letting

$$r_C(i) := \max\{d(i,j) | \ (i,j) \in \mathcal{E}_C\}.$$

[1]Here we consider only the energy required for transmission, thus not considering the energy required for reception or signal processing; these assumptions are the same as other well-studied models [12, 19, 3] and do not consider the joint contribution of different forms of energy expenditure and associated trade-offs.

[2]Obviously, the concept of 'real instance' is necessarily somewhat vacuous and hard to define mathematically.

Algorithm A_{MST}
Input: set of stations $\mathcal{S}$ on a k-dimensional Euclidean space, a source node $s \in \mathcal{S}$, and destination nodes $D \subseteq \mathcal{S}$;

1. Let $\mathcal{G} = (\mathcal{S}, \mathcal{E}, w)$ be the undirected weighted graph with $\mathcal{E} = \mathcal{S} \times \mathcal{S}$ and $w(i,j) = d(i,j)^\alpha$;
2. Compute a MST T of $\mathcal{G}$;
3. Compute a directed tree T^{dir} obtained as the union of all paths in T connecting s to some $d \in D$;
4. Return T^{dir}.

Figure 2: The A_{MST} algorithm.

Notice that $r_C(\cdot)$ is the minimum-cost range assignment such that $G_{r_C} = C$. We thus define $\mathsf{COST}(C) := \mathsf{COST}(r_C)$. So, the cost function $\mathsf{COST}(\cdot)$ measures the overall power consumption required by the network to implement a given communication graph.

Consider a communication graph C connecting a given source node $s \in \mathcal{S}$ to a set of *destination nodes* $D(C)$. Observe that every communication graph C can be transformed into a tree T such that $D(C) = D(T)$ and $\mathsf{COST}(T) \leq \mathsf{COST}(C)$. We can thus restrict ourselves to trees without any loss of efficiency. In particular, a *broadcast tree* is a tree $T = (\mathcal{S}, \mathcal{E}_T)$ such that $D(T) = \mathcal{S}$. Similarly, a *unicast tree* is a tree $P = (\mathcal{S}, \mathcal{E}_P)$ connecting a given source node s to a given destination node d. Clearly, an optimal unicast tree can be computed by considering the minimum-cost path connecting s to d, among all possible paths from s to d.

Given a set of stations $\mathcal{S}$, a source node s and a set of destination nodes $D \subseteq \mathcal{S}$, let $T^*(\mathcal{S}, s, D)$ denote the multicast tree connecting s to D of minimal cost. We denote by $\mathsf{OPT}(\mathcal{S}, s, D)$ its cost. An algorithm is an r-approximation algorithm if, for *any* input $(\mathcal{S}, s, D)$, (i) it runs in polynomial time and (ii) the computed multicast tree T satisfies $\mathsf{COST}(T) \leq r \cdot \mathsf{OPT}(\mathcal{S}, s, D)$.

1.2 Previous Related Work

Several algorithms constructing multicast trees in wireless networks have been proposed in the literature. One of the most natural algorithms is probably A_{MST} shown in Fig. 2 [12]. Essentially, A_{MST} constructs a MST on the set of points corresponding to the physical locations of the stations and then direct the edges from s to all other nodes that need to be reached (or that provide a connection from s to some $d \in D$). The A_{SPT} algorithm instead, constructs a shortest-path tree, rooted at the source node, of the complete graph $\mathcal{G}$ with edge weights $w(i,j) = d(i,j)^\alpha$. Both algorithms have been presented and experimentally evaluated in [12], while [5, 19] provide a worst-case analysis of them. In the remaining of this section we describe more in detail these and other related theoretical/experimental results.

1.2.0.1 Theoretical Results.

Assigning transmission powers to the stations which (i) guarantee a "good" communication graph, and (ii) minimize the overall power consumption of the network gives rise to interesting algorithmic questions. In particular, these two aspects yield a class of fundamental optimization problems, denoted as *range assignment* problems, that have been the subject of several works in the area of wireless network theory [15, 11, 10, 5, 19, 7, 6, 2, 3, 1, 18] (see also [8] for a survey). These works mainly focus on the existence of polynomial-time algorithms having a good *worst case* approximation guarantee.

In [5] it is proved that, when requiring the solution C to be a broadcast tree, the problem is NP-hard when stations are located in the 2-dimensional Euclidean space and for any $\alpha > 1$. For $\alpha = 1$, the problem admits a trivial optimal solution: directly connect s to the farthest station (and thus to all stations in $\mathcal{S}$). In [5, 19] it is proved that, when stations are located on a k-dimensional Euclidean space, and $\alpha \geq k$, the A_{MST} algorithm is a c_k^α-approximation, where $c_2^2 \leq 12$ in [19]. The proof of these results is based on the following bound on the total weight of a minimum spanning tree:

$$\mathsf{MST}(\mathcal{S}) \leq c_k^\alpha \cdot \mathsf{OPT}(\mathcal{S}, s), \tag{2}$$

where $\mathsf{OPT}(\mathcal{S}, s)$ denotes the minimal energy required by any broadcast tree for the instance $(\mathcal{S}, s)$, and $\mathsf{MST}(\mathcal{S}) := \sum_{(i,j)\in T_{MST}} d(i,j)^\alpha$, with T_{MST} being a minimum spanning tree of the point set $\mathcal{S} \subseteq \mathbf{R}^k$. In [5] it is proved that the approximation factor c_k^α increases exponentially with k, while [19] showed that $c_2^2 \geq 6$. Notice that, for $\alpha < k$, no constant-approximation algorithm is known. In this case, A_{MST} returns a solution of cost $\Omega(\sqrt{n})$ times the optimum even for very regular instances (e.g., stations forming a square grid). In [19] the authors proved that A_{SPT} has an unbounded approximation ratio, that is, for every $R > 1$, there exists an instance for which A_{SPT} returns a broadcast tree whose cost is more than R times the optimum.

The work [6] focuses on one-dimensional Euclidean instances and shows that a polynomial-time exact algorithm exists even when imposing that the broadcast tree must have depth at most h, for every given $h \geq 1$. Recently, [1] provide an exact polynomial-time algorithm for the 2-dimensional Euclidean case with $h = 2$, and a family of $(1+\epsilon)$-approximate algorithms for 2-dimensional instances and constant h, for any $\epsilon > 0$.

1.2.0.2 Experimental Results.

One of the main difficulties in experimentally evaluating the performances of a given algorithm is the fact that, since the problem is NP-hard [5], no polynomial-time algorithm for computing the optimum is known. Therefore, the possibilities are to (i) consider small-size instances for which non polynomial-time algorithms return the optimum in a reasonable amount of time, (ii) compare the algorithm at hand with other algorithms, or (iii) use a "sufficiently good" estimation of the optimum.

The authors of [12] propose three algorithms for this problem and experimentally evaluate their performances on 2-dimensional random instances generated with uniform distribution. In particular, two of the algorithms considered in [12] are A_{MST} and A_{SPT}. The experiments are performed on 100 instances generated for each of the following values: $|\mathcal{S}| \in \{10, 100\}$, $\alpha \in \{2, 4\}$ and several values of $|D|$. Not surprisingly, when $|D|$ gets larger, A_{SPT} gets worse than A_{MST}. In addition, the Multicast Incremental Power (MIP) algorithm proposed in the same work (a variant of Prim's MST algorithm) denotes a better performance of both algorithms above, when $|D|$ is sufficiently large w.r.t. $|\mathcal{S}|$. On the contrary, when $|D|$ is small, A_{SPT} yields better solutions that the other two algorithms. The computed solutions are evaluated w.r.t. the optimum only for small values of $|\mathcal{S}|$ (namely, $|\mathcal{S}| \leq 10$). For larger values, instead, each of the three algorithms is compared with the best solution returned by the other two (thus an estimation of the optimum).

In order to get around the problem of computing the optimum of an (randomly-generated) instance, the work [9] uses a different (geometric) approach for the case of broadcast trees. Given a minimum spanning tree T of a set of points $\mathcal{S}$, let $\mathsf{MST}(\mathcal{S}) := \sum_{(i,j)\in T} d(i,j)^\alpha$. Then, using the technique of [5, 19], it is possible to show that the approximation ration c_2^α of $\mathrm{A}_{\mathrm{MST}}$ is at most $4R(\alpha)$, where $R(\alpha) = \max_{\mathcal{S}\subseteq\mathbf{R}^2} \mathsf{MST}(\mathcal{S})/\mathcal{D}(\mathcal{S})^\alpha$, and $\mathcal{D}(\mathcal{S})$ is the diameter of the smallest disk containing $\mathcal{S}$. The best known theoretical upper bound is $R(2) \leq 3$ [19], thus a worst-case approximation factor 12 for $\mathrm{A}_{\mathrm{MST}}$ when $\alpha = 2$. However, the experiments in [9] show that, for random instances, this value is much smaller. Indeed, for several network sizes, 10,000 instances were generated and, in all such cases, $4\mathsf{MST}(\mathcal{S})/\mathcal{D}(\mathcal{S})^\alpha$ was less than 6.4. Interestingly, the highest values were all achieved for small-size instances (i.e., $|\mathcal{S}| \leq 9$), while $\mathsf{MST}(\mathcal{S})/\mathcal{D}(\mathcal{S})^\alpha$ decreases as $|\mathcal{S}|$ increases.

It is worth observing that the approximation ratio of $\mathrm{A}_{\mathrm{MST}}$ is an important parameter also for other algorithms. For instance, in [19] it is shown that, for broadcast trees, the approximation ratio of algorithm MIP proposed in [12] is at most 12 by showing that the cost of the latter is not larger than the cost of $\mathrm{A}_{\mathrm{MST}}$.

1.3 Our Contribution

In this work we experimentally evaluate a new multicast algorithm based on the construction of *Light Approximate Shortest-path Trees* (LASTs). LASTs have been introduced in [14] where the authors proved that, for any parameter $\delta > 1$, for any weighted directed graph $\mathcal{G}$ and for any source node s in $\mathcal{G}$, it is possible to construct in polynomial time a $(\delta, 1 + \frac{2}{\delta-1})$-LAST, that is, a spanning tree T of $\mathcal{G}$ such that (see [14, Theorem 4]):

- In T the length of the path from s to every other node d is at most δ times the length of the path from s to d in the shortest-path tree of $\mathcal{G}$ rooted at s (i.e., T approximates the shortest-path tree distances by a factor δ);

- The sum of the weights of the edges in T is at most $1+\frac{2}{\delta-1}$ times the cost of the MST of $\mathcal{G}$ (i.e., T approximates the cost of the MST by a factor $1 + \frac{2}{\delta-1}$).

Intuitively speaking, the parameter δ provides a continuous trade-off between MSTs and SPTs: the larger δ the closer the cost to weight of a MST; the smaller δ, the closer are the distances to the minimal distances in $\mathcal{G}$.

The idea of using LASTs for multicast in wireless networks has been first introduced in a work from the same authors [18], motivated by applications in a game-theoretic setting involving selfishly-acting users (i.e., receivers of a transmission located nearby the stations). The resulting algorithm $\mathrm{A}_{\mathrm{LAST}}$ is obtained by replacing, in Step 2 of $\mathrm{A}_{\mathrm{MST}}$, the MST algorithm with the one in [14] (the latter is based on the idea of replacing parts of a MST by parts of a SPT). By combining the results on $\mathrm{A}_{\mathrm{MST}}$ in [19] (see Eq. 2 above) with the above mentioned properties of LASTs, it is easy to prove the following bounds on the *worst case* approximation guarantee of $\mathrm{A}_{\mathrm{LAST}}$:

$$\frac{\mathsf{COST}(\mathrm{A}_{\mathrm{LAST}}(\mathcal{S},s,D))}{\mathsf{OPT}(\mathcal{S},s,D)} \leq \begin{cases} \left(1+\frac{2}{\delta-1}\right)\frac{\mathsf{MST}(\mathcal{S})}{\mathsf{OPT}(\mathcal{S},s,D)} \leq 12 + \frac{24}{\delta-1} & \text{if } D = \mathcal{S}, \\ \delta \cdot |D| & \text{otherwise.} \end{cases} \tag{3}$$

So, the theoretical bounds above say that, when considering broadcast trees, LASTs have only a constant factor loss of efficiency w.r.t. that of MSTs (e.g., when $\delta = 2$, we are only guaranteed that $\mathrm{A}_{\mathrm{LAST}}$ returns a solution of cost at most three times $\mathsf{MST}(\mathrm{S})$, thus an approximation factor 36). It is worth observing that, denoted $\lambda_\delta := 1 + \frac{2}{\delta-1}$, the above mentioned bounds do *not* imply that $\mathsf{COST}(\mathrm{A}_{\mathrm{LAST}}(\mathcal{S},s)) \leq \lambda_\delta \cdot \mathsf{COST}(\mathrm{A}_{\mathrm{MST}}(\mathcal{S},s))$; all we can obtain from these bounds is the following (much weaker) one:

$$\mathsf{COST}(\mathrm{A}_{\mathrm{LAST}}(\mathcal{S},s)) \leq 12\lambda_\delta \cdot \mathsf{COST}(\mathrm{A}_{\mathrm{MST}}(\mathcal{S},s)). \tag{4}$$

We experimentally evaluate the loss of efficiency obtained when replacing a MST with a LAST, that is, how much we have to increase the cost of broadcasting if we also want to approximate the cost of *every* single source-destination path (i.e., the cost of unicast) within a factor δ. In particular, we consider how often $\mathrm{A}_{\mathrm{LAST}}$ worsen the solution of $\mathrm{A}_{\mathrm{MST}}$ and, if so, how far are the two costs. We evaluate these parameters for instances of $n = |\mathcal{S}|$ nodes generated at random with uniform distribution, for several values of n, δ and α. Fixed these parameters, we generate several thousands of instances by throwing n points at random with uniform probability (i.e., the position of the stations) and then choosing the source with uniform distribution. Our results show that the loss of efficiency is well-below a factor λ_δ, that is, in all cases

$$\mathsf{COST}(\mathrm{A}_{\mathrm{LAST}}(\mathcal{S},s)) \leq \lambda_\delta^{exp} \cdot \mathsf{COST}(\mathrm{A}_{\mathrm{MST}}(\mathcal{S},s)),$$

with $\lambda_\delta^{exp} < \lambda_\delta$ (e.g., for $\delta = 2$ and $\alpha = 2$ we obtain an experimental upper bound $\lambda_\delta^{exp} = 1.572$). So, experimentally, the bound is much below the one in Eq. 4 (actually even below λ_δ itself).

For each randomly-generated instance $(\mathcal{S}, s)$ we consider the following ratios:

$$\begin{aligned} \lambda_\delta^{exp}(\mathcal{S},s) &:= \frac{\mathsf{COST}(\mathrm{A}_{\mathrm{LAST}}(\mathcal{S},s))}{\mathsf{COST}(\mathrm{A}_{\mathrm{MST}}(\mathcal{S},s))}, \\ \sigma_\delta^{exp}(\mathcal{S},s) &:= \frac{\mathsf{COST}(\mathrm{A}_{\mathrm{SPT}}(\mathcal{S},s))}{\mathsf{COST}(\mathrm{A}_{\mathrm{MST}}(\mathcal{S},s))}, \\ \overline{\lambda}_\delta^{exp}(\mathcal{S},s) &:= \frac{\mathsf{COST}(\mathrm{A}_{\mathrm{LAST}}(\mathcal{S},s))}{\mathsf{MST}(\mathcal{S})}, \\ \overline{\sigma}_\delta^{exp}(\mathcal{S},s) &:= \frac{\mathsf{COST}(\mathrm{A}_{\mathrm{SPT}}(\mathcal{S},s))}{\mathsf{MST}(\mathcal{S})}. \end{aligned}$$

Clearly, the worst case (i.e., the largest value obtained in our experiments) of λ_δ^{exp} and σ^{exp} show the worst behavior, for randomly-generated instances, of $\mathrm{A}_{\mathrm{LAST}}$ and $\mathrm{A}_{\mathrm{SPT}}$ w.r.t. $\mathrm{A}_{\mathrm{MST}}$. In addition, we relate the above ratios to the approximation ratio of $\mathrm{A}_{\mathrm{MST}}$ and provide a *twofold* connection. First, the best case $\lambda_\delta^{exp}(\mathcal{S},s)$ provides a *lower bound* on the approximation ratio achieved by $\mathrm{A}_{\mathrm{MST}}$ on our randomly-generated instances. Indeed, if $\lambda_\delta^{exp}(\mathcal{S}_{best}, s_{best}) = 1/c$ then, for some instance $(\mathcal{S}_{best}, s_{best})$, it holds that

$$\mathsf{COST}(\mathrm{A}_{\mathrm{MST}}(\mathcal{S}_{best},s_{best})) = c \cdot \mathsf{COST}(\mathrm{A}_{\mathrm{LAST}}(\mathcal{S}_{best},s_{best}))$$

and thus the cost of $\mathrm{A}_{\mathrm{MST}}(\mathcal{S}_{best}, s_{best})$ is at least c times the optimum on that instance (the same holds when considering the best case of $\sigma_{\delta}^{exp}(\mathcal{S}, s)$). This compares with the experimental *upper bound* in [9] mentioned above. On the other hand, the weight $\mathsf{MST}(\mathcal{S})$ of the minimum spanning tree of $\mathcal{S}$ is related to the optimum via the relation in Eq. 2. We thus provide an absolute *upper bound* on the approximation ratio achieved by $\mathrm{A}_{\mathrm{LAST}}$ and $\mathrm{A}_{\mathrm{SPT}}$ by considering the worst case of $\overline{\lambda}_{\delta}^{exp}$ and $\overline{\sigma}_{\delta}^{exp}$ (see Sect. 2.4). The results on these bounds parallel with the results on the worst case of λ_{δ}^{exp} and σ_{δ}^{exp}: we indeed obtain

$$\mathsf{COST}(\mathrm{A}_{\mathrm{LAST}}(\mathcal{S}, s)) \leq 12\lambda_{\delta}^{exp} \cdot \mathsf{OPT}(\mathcal{S}, s), \qquad (5)$$

where, for $\delta = 2$, $12 \cdot \lambda_{\delta}^{exp} < 18.864 < 12\lambda_{\delta} = 36$.

By contrast, the theoretical bound δ on the approximation ratio of the unicast is tight also for randomly-generated instances (see Sect. 2.2). Therefore, the goal is to find the smallest δ for which $\mathrm{A}_{\mathrm{LAST}}$ provides a sufficiently good broadcast tree (see Sect. 2.3). This also explains our initial choice $\delta = 2$ for several of our experiments: this value is a good "compromise" when considering the *maximum* between the approximation ratios for unicast and for broadcast (i.e., λ_{δ} and δ). Indeed, there are two "symmetric" cases: the case $\delta = 2$ and $\lambda_{\delta} = 3$, and the case $\delta = 3$ and $\lambda_{\delta} = 2$. Since the bound δ is experimentally tight, while $\lambda_{\delta}^{exp} < \lambda_{\delta}$, it makes no sense to consider any $\delta > 2$ (in particular, the case $\delta = 3$).

1.3.0.3 Paper organization.

In Sect. 2 we present our experimental results. In particular, in Sect.s 2.1 and 2.1.1, we consider the cost of broadcast trees for $\delta = 2$ and $\alpha = 2, 4, 8$: Sect. 2.1 deals with $|\mathcal{S}| \geq 10$, while Sect. 2.1.1 considers the case $|\mathcal{S}| \leq 10$. The experimental results on the cost of unicast are contained in Sect. 2.2. In Sect. 2.3 we consider different values of δ and evaluate λ_{δ}^{exp}. The results on $\overline{\lambda}_{\delta}^{exp}$ and $\overline{\sigma}_{\delta}^{exp}$ are contained in Sect. 2.4. In Sect. 3 we conclude and outline some future research and open questions.

2. EXPERIMENTAL RESULTS

We first compare both $\mathrm{A}_{\mathrm{LAST}}$ and $\mathrm{A}_{\mathrm{SPT}}$ to $\mathrm{A}_{\mathrm{MST}}$ and consider the worst and the best cases of the first two algorithms, i.e., the instances for which $\mathrm{A}_{\mathrm{LAST}}$ and $\mathrm{A}_{\mathrm{SPT}}$ performed worst/best w.r.t. $\mathrm{A}_{\mathrm{MST}}$. We start by looking at the case $\delta = 2$ and $\alpha = 2$ and obtain the following results for $\alpha = 2$ and $\delta = 2$:

- For small-size instances (i.e., $5 \leq |\mathcal{S}| \leq 10$) in at least 95.230% of the cases the solution computed by $\mathrm{A}_{\mathrm{LAST}}$ in not worse than that of $\mathrm{A}_{\mathrm{MST}}$. Moreover, the solution computed by $\mathrm{A}_{\mathrm{LAST}}$ is better than that of $\mathrm{A}_{\mathrm{MST}}$ in at least 1.238% of the cases. For larger size instances (i.e., $|\mathcal{S}| = 10, 20, \ldots, 50, 100, 150, 200$) $\mathrm{A}_{\mathrm{LAST}}$ computes a solution better (resp., not worse) than $\mathrm{A}_{\mathrm{MST}}$ in at lest 4.190% (resp., 5.330%) of the cases; these values are achieved for $|\mathcal{S}| = 200$ and the percentages get better as $|\mathcal{S}|$ gets smaller. In that respect, $\mathrm{A}_{\mathrm{SPT}}$ has a much worse behavior, since for $|\mathcal{S}| \leq 40$ it returns a solution not worse than $\mathrm{A}_{\mathrm{MST}}$ in only 6.57% of the cases. This percentage drops down to 0 for $|\mathcal{S}| \in \{150, 200\}$, where $\mathrm{A}_{\mathrm{SPT}}$ always returns a solution worse than $\mathrm{A}_{\mathrm{MST}}$.
- Although for large-size instances there is a quite large fraction of the instances for which $\mathrm{A}_{\mathrm{LAST}}$ worsen the solution computed by $\mathrm{A}_{\mathrm{MST}}$, in *all* such cases, the solution of $\mathrm{A}_{\mathrm{LAST}}$ is not much worse than that of $\mathrm{A}_{\mathrm{MST}}$. Indeed, in the worst case, the ratio $\lambda_{\delta}^{exp}(\mathcal{S}, s)$ is 1.572. This value is achieved for small size instances, while large-size instances yield $\lambda_{\delta}^{exp}(\mathcal{S}, s) \leq 1.463$. On the contrary, the worst ratio $\sigma_{\delta}^{exp}(\mathcal{S}, s)$ is always greater than the corresponding ratio $\lambda_{\delta}^{exp}(\mathcal{S}, s)$. Moreover, the ratio $\sigma_{\delta}^{exp}(\mathcal{S}, s)$ has maximum value 2.493 and, all worst cases are greater than 1.59.
- When considering the minimal ratios $\lambda_{\delta}^{exp}(\mathcal{S}, s)$ and $\sigma_{\delta}^{exp}(\mathcal{S}, s)$ achieved in our experiments (i.e., the best case), we have the following results. $\mathrm{A}_{\mathrm{LAST}}$ is better for $|\mathcal{S}| \geq 50$, while $\mathrm{A}_{\mathrm{SPT}}$ is better for $|\mathcal{S}| \leq 40$. For all instances sizes, in the best case, $\lambda_{\delta}^{exp}(\mathcal{S}, s) < 1$, while $\sigma_{\delta}^{exp}(\mathcal{S}, s) > 1$ for $|\mathcal{S}| \in \{150, 200\}$.

By looking closer to the above three aspects one can see that the good performances of $\mathrm{A}_{\mathrm{LAST}}$ are not only due to the fact that for some network parameters (i.e., $|\mathcal{S}|$ and α) $\mathrm{A}_{\mathrm{SPT}}$ performs better than $\mathrm{A}_{\mathrm{MST}}$: indeed, $\mathrm{A}_{\mathrm{LAST}}$ is better than $\mathrm{A}_{\mathrm{MST}}$ even for those cases in which $\mathrm{A}_{\mathrm{SPT}}$ is not better than $\mathrm{A}_{\mathrm{MST}}$ (see, in Table 1, the cases $|\mathcal{S}| = 100, 150$). In other words, the "compromise" between minimum spanning trees and shortest-path trees provided by LASTs is better than simply considering the best solution of the two algorithms $\mathrm{A}_{\mathrm{MST}}$ and $\mathrm{A}_{\mathrm{SPT}}$. For instance, for $|\mathcal{S}| \geq 50$ and $\alpha = 2, 4$, there are instances for which $\mathrm{A}_{\mathrm{LAST}}$ is better than both $\mathrm{A}_{\mathrm{MST}}$ and $\mathrm{A}_{\mathrm{SPT}}$ (see best cases in Table 1 and further results in [17]).

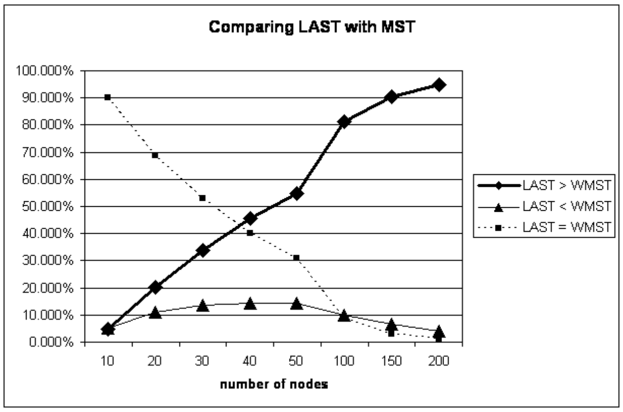
2.1 Cost of broadcast

In this section we evaluate the relative performance of algorithm $\mathrm{A}_{\mathrm{LAST}}$ versus $\mathrm{A}_{\mathrm{MST}}$ and $\mathrm{A}_{\mathrm{SPT}}$. We first count how many times $\mathrm{A}_{\mathrm{LAST}}$ and $\mathrm{A}_{\mathrm{SPT}}$ yield a solution better/worse than $\mathrm{A}_{\mathrm{MST}}$. The results are shown in Fig. 3 for $\alpha = 2$ and $\delta = 2$. Observe that $\mathrm{A}_{\mathrm{SPT}}$ degrades quite fast w.r.t. $\mathrm{A}_{\mathrm{MST}}$ as the number of nodes increase. On the contrary, $\mathrm{A}_{\mathrm{LAST}}$ has a different behavior since there is a non-zero fraction of the instances for which it returns a solution better than $\mathrm{A}_{\mathrm{MST}}$.

In Table 1 we report on the worst/best ratio achieved by $\mathrm{A}_{\mathrm{LAST}}$ and $\mathrm{A}_{\mathrm{SPT}}$ w.r.t. the solution yield by $\mathrm{A}_{\mathrm{MST}}$. It is evident the improvement of $\mathrm{A}_{\mathrm{LAST}}$ over $\mathrm{A}_{\mathrm{SPT}}$ in the worst case. More importantly, in the worst case, the cost of solution $\mathrm{A}_{\mathrm{LAST}}(\mathcal{S}, s)$ is well below $\lambda_{\delta}\mathsf{COST}(\mathrm{A}_{\mathrm{MST}}(\mathcal{S}, s)) = 3\mathsf{COST}(\mathrm{A}_{\mathrm{MST}}(\mathcal{S}, s))$.

When considering the best achieved ratio, we notice that the $\mathrm{A}_{\mathrm{SPT}}$ is better for $\mathcal{S} \leq 40$ while $\mathrm{A}_{\mathrm{LAST}}$ is better for larger size instances. We recall that the analysis of the best case gives a useful insight on the approximation ratio of $\mathrm{A}_{\mathrm{MST}}$ for instances generated with uniform distribution (see the discussion in Sect. 1.3). Indeed, by considering the best of $\mathrm{A}_{\mathrm{SPT}}$ and $\mathrm{A}_{\mathrm{LAST}}$ as an upper bound to the optimum, we can derive a *lower bound* on the approximation ratio of $\mathrm{A}_{\mathrm{MST}}$. In the third column of Table 1 we show such a lower bound for the values considered.

We next consider attenuation parameter $\alpha = 4$ (as in [12]) and $\alpha = 8$. We compare to the experiments for the case $\alpha = 2$. Observe that, for $\alpha = 4, 8$, there is a higher number of instances in which $\mathrm{A}_{\mathrm{LAST}}$ and $\mathrm{A}_{\mathrm{MST}}$ provide a solution with the same cost (see [17]). This can be explained with



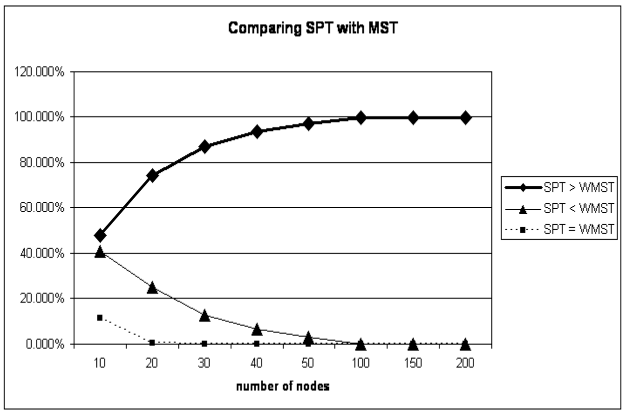


Figure 3: The frequency of solutions returned by $\mathrm{A}_{\mathrm{LAST}}$ and $\mathrm{A}_{\mathrm{SPT}}$ which are better/as good as/worse than the solution of $\mathrm{A}_{\mathrm{MST}}$ (10,000 randomly-generated instances with $\alpha = 2$ and $\delta = 2$).

the fact that $\mathrm{A}_{\mathrm{LAST}}$ builds its tree starting from the MST. Therefore, as α increase, $\mathrm{A}_{\mathrm{LAST}}$ returns the same solution as $\mathrm{A}_{\mathrm{MST}}$ in a larger number of cases. More importantly, $\mathrm{A}_{\mathrm{LAST}}$ maintains its performances for $\alpha = 4, 8$, while in general the worst case of $\mathrm{A}_{\mathrm{SPT}}$ gets higher.

Our data, for $\alpha = 2$ (and $|\mathcal{S}| \geq 50$), provide a valid justification in the use of LASTs for constructing broadcast trees, as opposed to the simpler/natural approach of taking the best solution returned by $\mathrm{A}_{\mathrm{MST}}$ and $\mathrm{A}_{\mathrm{SPT}}$. This also holds for $\alpha = 4$ and $|\mathcal{S}| \geq 50$ (see [17]).

2.1.1 A closer look to small-size instances

In this section we focus on the study of small-size instances, since these cases seem to be the most promising for discovering significant improvements of $\mathrm{A}_{\mathrm{LAST}}$ and of $\mathrm{A}_{\mathrm{SPT}}$ in the worst/best case. As previously mentioned, the best case yield a lower bound on the approximation ratio of $\mathrm{A}_{\mathrm{MST}}$, that is, what is the worst ratio achieved by $\mathrm{A}_{\mathrm{MST}}$ over all instances generated. In particular, for network sizes $5 \leq |\mathcal{S}| \leq 9$ we generate 50,000 instances and obtain the results in Table 2.

Observe that we obtain an higher worst ratio for $\mathrm{A}_{\mathrm{LAST}}$ w.r.t. the large-size instances (see Table 1). This fact shows that small-size instances are a severe test for the three algorithms (thus a similar trend of the upper bound for $\mathrm{A}_{\mathrm{MST}}$ in [2]). However, also in these cases, the value $\lambda_\delta^{exp}(\mathcal{S}, s)$ is well below the theoretical bound $\lambda_2 = 3$.

2.2 Cost of unicast

So far we have been evaluating the cost of the broadcast tree obtained from the three algorithms $\mathrm{A}_{\mathrm{MST}}$ $\mathrm{A}_{\mathrm{SPT}}$ and $\mathrm{A}_{\mathrm{LAST}}$. We next consider the experimental worst case for the cost of unicast produced by the same three algorithms. The obtained results are shown in Table 3 and 4.

Notice that the theoretical bound matches with the experimental one (see the "Worst case" column of Tables 3 and 4). More importantly, $\mathrm{A}_{\mathrm{MST}}$ has a much worse behavior w.r.t. the $\mathrm{A}_{\mathrm{LAST}}$ in both best and worse case. Moreover, for small-size instances, $\mathrm{A}_{\mathrm{LAST}}$ achieves the optimal unicast in a good percentage of cases. In this analysis $\mathrm{A}_{\mathrm{MST}}$ is also below $\mathrm{A}_{\mathrm{LAST}}$. Our data confirms the intuition that MST is not a good unicast tree, while LASTs give a δ-approximation of the optimal unicast. We also tested other values of $\alpha > 2$ and obtained analogous tight results.

2.3 Adjusting the parameter δ of the LASTs

We have seen that the theoretical unicast bound δ matches with the experimental one δ^{exp} (see Sect. 2.2), while $\lambda_\delta^{exp} < \lambda_\delta$ (see Sect. 2.1). The idea is the following: try to adjust δ to obtain a better unicast tree with a "good" λ_δ^{exp}. Our trials are shown in Table 5.

Our experiments give a slightly higher λ_δ^{exp} w.r.t. the cases in which $\delta = 2$. We can see that, in these cases, the differences between λ_δ^{exp} and λ_δ are quite important (see e.g. the experimental bound for $\lambda_{1.01} = 201$). So, at the price of a small degradation of the broadcast tree, we obtain a much better unicast approximation.

2.4 Cost of broadcast versus weight of MSTs

In this section we consider the ratios $\overline{\lambda}_\delta^{exp}$ and $\overline{\sigma}_\delta^{exp}$. As mentioned in the Introduction, these ratios give an absolute upper bound on the approximation ratio achieved by $\mathrm{A}_{\mathrm{LAST}}$ and $\mathrm{A}_{\mathrm{SPT}}$, respectively. The results are shown in Table 6.

	Worst ratio		Best ratio	
Nodes	$\overline{\lambda}_\delta^{exp}$	$\overline{\sigma}_\delta^{exp}$	$\overline{\lambda}_\delta^{exp}$	$\overline{\sigma}_\delta^{exp}$
10	**1.393484**	2.959805	**0.365746**	**0.361863**
20	1.334686	**3.016235**	0.510485	0.545921
30	1.267987	2.690986	0.634819	0.739471
40	1.325315	2.60085	0.675511	0.840242
50	1.21736	2.440541	0.703407	0.841885
100	1.130818	2.396243	0.772127	1.123973
150	1.138154	2.553272	0.785942	1.269689

Table 6: The worst/best ratio between the cost of the solutions of $\mathrm{A}_{\mathrm{LAST}}$ and $\mathrm{A}_{\mathrm{SPT}}$ over MST (10,000 randomly-generated instances with $\alpha = 2$ and $\delta = 2$).

3. CONCLUSIONS AND FUTURE WORK

The theoretical (upper and lower) bounds on the approximation ratio of $\mathrm{A}_{\mathrm{MST}}$ and $\mathrm{A}_{\mathrm{LAST}}$ are not tight, especially when looking at randomly-generated instances. This is due to the fact that the cost is usually upper bounded by summing up the weight of *all* edges in the solution. This explains the fact that, in some cases, $\mathrm{A}_{\mathrm{MST}}$ is worse than $\mathrm{A}_{\mathrm{LAST}}$ and/or $\mathrm{A}_{\mathrm{SPT}}$: though $\mathrm{A}_{\mathrm{LAST}}$ increases the total edge weights of the MST, we not always have to pay for all such edges (see, e.g., the instance in Fig. 4).

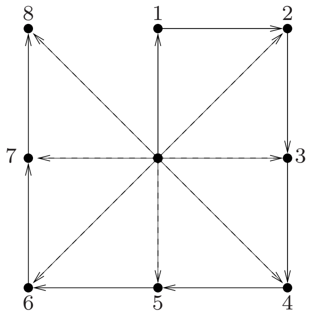


Figure 4: An instance for which A_{LAST} and A_{SPT} yield a solution of cost $1/4$ the cost of the solution of A_{MST}, for $\alpha = 2$ (dashed edges belongs to a SPT, while solid ones belong to a MST).

An interesting open problem is to investigate lower bounds on the approximation ratio of A_{LAST}. In particular, given that A_{MST} is at least 6-approximate [19], is there an instance for which $\mathsf{COST}(A_{LAST}(\mathcal{S}, s)) > 6\lambda_\delta \mathsf{OPT}(\mathcal{S}, s)$? As for the upper bound, it would be interesting to see whether $\mathsf{COST}(A_{LAST}(\mathcal{S}, s)) < \lambda_\delta \mathsf{COST}(A_{MST}(\mathcal{S}, s))$. Finally, the existence of a polynomial-time constant approximation algorithm for the multicast problem, i.e., when a $D \subset \mathcal{S}$ is given, is open.

4. REFERENCES

[1] C. Ambuehl, A. Clementi, M. Di Ianni, N. Lev-Tov, A. Monti, D. Peleg, G. Rossi, and R. Silvestri. Efficient algorithms for low-energy bounded-hop broadcast in ad-hoc wireless networks. In *Proc. of Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, LNCS 2996, pages 418–427, 2004.

[2] C. Ambuehl, A. Clementi, P. Penna, G. Rossi, and R. Silvestri. Energy Consumption in Radio Networks: Selfish Agents and Rewarding Mechanisms. In *Proceedings of the 10th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 1–16, 2003.

[3] L. Anderegg and S. Eidenbenz. Ad hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents. In *Proc. of the Annual ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2003.

[4] A. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca. A Worst-case Analysis of a MST-based Heuristic to Construct Energy-efficient Broadcast Subtrees in Wireless Networks. TR 010, Univ. of Rome "Tor Vergata", http://www.mat.uniroma2.it/~penna/papers/stacs01-TR.ps.gz, 2001.

[5] A. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca. On the complexity of computing minimum energy consumption broadcast subgraphs. *Proc. of the Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, LNCS(2010):121–131, 2001. Full version in [4].

[6] A. Clementi, M. Di Ianni, and R. Silvestri. The minimum broadcast range assignment problem on linear multi-hop wireless networks. *Theoretical Computer Science*, Theor. Comput. Sci. 1-3(299): 751-761.

[7] A. Clementi, A. Ferreira, P. Penna, S. Perennes, and R. Silvestri. The minimum range assignment problem on linear radio networks. *Algorithmica*, 35:95–110, 2003.

[8] A. Clementi, G. Huiban, P. Penna, G. Rossi, and Y.C. Verhoeven. Some recent theoretical advances and open questions on energy consumption in ad-hoc wireless networks. In *Proceedings of the 3rd Workshop on Approximation and Randomization Algorithms in Communication Networks (ARACNE)*, pages 23–38, 2001.

[9] A. Clementi, G. Huiban, P. Penna, G. Rossi, and Y.C. Verhoeven. On the Approximation Ratio of the MST-based Heuristic for the Energy-Efficient Broadcast Problem in Static Ad-Hoc Radio Networks. In *In Proc. of the 1st Workshop on Wireles, Mobile and Ad-Hoc Networks (WMAN)*, 2003.

[10] A. Clementi, P. Penna, and R. Silvestri. The power range assignment problem in radio networks on the plane. In *Proc. of the 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, number 1770 in LNCS, pages 651–660, 2000.

[11] A. Clementi, P. Penna, and R. Silvestri. The Power Assignment Problem in Radio Networks. *ACM Mobile Networks and Applications*, 9(2):125–140, 2004.

[12] A. Ephremides, G.D. Nguyen, and J.E. Wieselthier. On the Construction of Energy-Effficient Broadcast and Multicast Trees in Wireless Networks. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 585–594, 2000.

[13] M.R. Garey and D.S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness.* Freeman, 1979.

[14] S. Khuller, B. Raghavachari, and N. Young. Balancing minimum spanning trees and shortest-path trees. *Algorithmica*, 14(4):305–321, 1995.

[15] L.M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *Theoretical Computer Science*, 243:289–305, 2000. Extended abstract in Proc. of STACS'97.

[16] K. Pahlavan and A. Levesque. *Wireless information networks.* Wiley-Interscience, 1995.

[17] P. Penna and C. Ventre. Energy-efficient broadcasting in ad-hoc networks: combining msts with shortest-path trees. Technical report, Technical Report of the European Project CRESCCO at www.ceid.upatras.gr/crescco, 2003.

[18] P. Penna and C. Ventre. Sharing the cost of multicast transmissions in wireless networks. In *Proc. of the International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, LNCS, 2004.

[19] P.-J. Wan, G. Calinescu, X.-Y. Li, and O. Frieder. Minimum-energy broadcasting in static ad hoc wireless networks. *Proc. of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2001. Journal version in *Wireless Networks*, 8(6): 607-617, 2002.

Nodes	Worst ratio $\lambda_\delta^{exp}(\mathcal{S},s)$	$\sigma_\delta^{exp}(\mathcal{S},s)$	Best ratio $\lambda_\delta^{exp}(\mathcal{S},s)$	$\sigma_\delta^{exp}(\mathcal{S},s)$	Lower bound on the apx ratio of $\mathrm{A_{MST}}$
10	1.45473675	**2.305245347**	**0.612590799**	**0.348877936**	**2.866332026**
20	1.441095633	1.967276868	0.743912924	0.61227423	1.633255086
30	**1.463335458**	2.144286978	0.803460083	0.66260896	1.50918575
40	1.437470634	2.040225473	0.824313711	0.778277883	1.284888113
50	1.455386458	1.928186937	0.852621393	0.853503816	1.172853517
100	1.259035589	1.749475918	0.927487388	0.980521707	1.078181777
150	1.243433947	1.624524465	0.955362459	1.026332578	1.046723147
200	1.22694007	1.598773982	0.96196238	1.068397079	1.039541691

Table 1: The worst/best ratio between the cost of the solutions of $\mathrm{A_{LAST}}$ and $\mathrm{A_{SPT}}$ over the cost of $\mathrm{A_{MST}}$ (10,000 randomly-generated instances with $\alpha = 2$ and $\delta = 2$).

Nodes	Worst ratio $\lambda_\delta^{exp}(\mathcal{S},s)$	$\sigma_\delta^{exp}(\mathcal{S},s)$	Best ratio $\lambda_\delta^{exp}(\mathcal{S},s)$	$\sigma_\delta^{exp}(\mathcal{S},s)$	Lower bound on the apx ratio of $\mathrm{A_{MST}}$
5	1.276516007	1.947219394	**0.536916526**	0.385731614	2.592476124
6	1.34547619	2.250435145	0.548756433	0.370500023	2.699055166
7	1.438073394	**2.492612128**	0.546624261	**0.352824185**	**2.83427283**
8	1.493178373	2.433231869	0.57799932	0.396503839	2.522043675
9	**1.571838125**	2.45137697	0.585923696	0.387225575	2.582474054
10	1.547802023	2.437001382	0.618862861	0.359872096	2.778765039

Table 2: The worst/best ratio between the cost of the solutions of $\mathrm{A_{LAST}}$ and $\mathrm{A_{SPT}}$ over the cost of $\mathrm{A_{MST}}$ (50,000 randomly-generated instances with $\alpha = 2$ and $\delta = 2$).

Nodes	worst case $\mathrm{A_{MST}}$	$\mathrm{A_{LAST}}$	best case $\mathrm{A_{MST}}$	$\mathrm{A_{LAST}}$	achieved opt $\mathrm{A_{MST}}$	$\mathrm{A_{LAST}}$
10	4.736082	2	1	1	10.50%	12.26%
20	7.012582	2	1	1	0.37%	0.73%
30	10.08234	2	1	1	0.01%	0.04%
40	11.29895	1.999703	1.03093	1.023071	0.00%	0.00%
50	14.53238	2	1.036549	1.036549	0.00%	0.00%
100	17.5929	2	1.24436	1.104489	0.00%	0.00%
150	20.67111	2	1.40402	1.209489	0.00%	0.00%
200	25.11876	2	1.476016	1.216234	0.00%	0.00%

Table 3: The worst/best ratio between the cost of unicast in $\mathrm{A_{MST}}$ and $\mathrm{A_{LAST}}$ (10,000 randomly-generated instances with $\alpha = 2$ and $\delta = 2$).

Nodes	worst case $\mathrm{A_{MST}}$	$\mathrm{A_{LAST}}$	best case $\mathrm{A_{MST}}$	$\mathrm{A_{LAST}}$	achieved opt $\mathrm{A_{MST}}$	$\mathrm{A_{LAST}}$
5	3.302363	1.999786	1	1	48.05%	48.83%
6	3.821315	1.999953	1	1	35.48%	36.78%
7	4.446378	2	1	1	26.33%	27.84%
8	5.035519	2	1	1	19.00%	20.64%
9	5.062921	2	1	1	13.86%	15.48%
10	5.662891	2	1	1	10.14%	11.79%

Table 4: The worst/best ratio between the cost of unicast in $\mathrm{A_{MST}}$ and $\mathrm{A_{LAST}}$ (50,000 randomly-generated instances with $\alpha = 2$ and $\delta = 2$).

δ	λ_δ	Nodes	Worst ratio $\lambda_\delta^{exp}(\mathcal{S},s)$	$\sigma_\delta^{exp}(\mathcal{S},s)$	Best ratio $\lambda_\delta^{exp}(\mathcal{S},s)$	$\sigma_\delta^{exp}(\mathcal{S},s)$
1.5	5	10	1.747268	2.610911	0.425091	0.387851
	5	100	1.376454	1.679471	0.929351	0.970396
1.25	9	10	1.825356	2.295495	0.425505	0.425505
	9	100	1.451658	1.751842	0.923907	0.969727
1.1	21	10	2.092178	2.338308	0.391455	0.376636
	21	100	1.538815	1.97884	0.954313	0.978234
1.05	41	10	2.097929	2.161458	0.365864	0.365864
	41	100	1.826472	1.806227	0.949624	0.965029
1.01	201	10	2.378215	2.378215	0.380838	0.380838
	201	100	1.699624	1.719947	0.97714	0.975886

Table 5: The worst/best ratio between the cost of broadcast for several values of $\delta < 2$ (50,000 randomly-generated instances of size 10 and 10,000 instances of size 100, with $\alpha = 2$).