An Approximation Algorithm for the Wireless Gathering Problem ⋆

Vincenzo Bonifaci [a,b,∗,1] Peter Korteweg [c,2]
Alberto Marchetti-Spaccamela [b,3] Leen Stougie [c,d,4]

[a] Dept. Electrical and Information Engineering, University of L'Aquila, Italy
[b] Dept. Computer and Systems Science, University of Rome "La Sapienza", Italy
[c] Dept. Mathematics and Computer Science, TU Eindhoven, The Netherlands
[d] CWI Amsterdam, The Netherlands

Abstract

The Wireless Gathering Problem is to find an interference-free schedule for data gathering in a wireless network in minimum time. We present a 4-approximate polynomial-time on-line algorithm for this **NP**-hard problem. We show that no shortest path following algorithm can have approximation ratio better than 4.

Key words: wireless networks, data gathering, approximation algorithms

⋆ A preliminary version of this paper has been published in the Proceedings of the 10th Scandinavian Workshop on Algorithms and Theory (SWAT), Lecture Notes in Computer Science, Volume 4059, pages 328-338, (2006).
∗ Corresponding author. Address: Dipartimento di Informatica e Sistemistica, Sapienza Università di Roma, Via Ariosto 25, 00185 Roma, Italy.
Email address: `bonifaci@dis.uniroma1.it` (Vincenzo Bonifaci).
[1] Research supported by EU COST-action 293 GRAAL and by EU FET Project ARRIVAL FP6-021235-2.
[2] Research supported by EU COST-action 293 GRAAL.
[3] Research supported by EU FET Integrated Project AEOLUS IST-15964 and by MIUR-FIRB Israel-Italy project RBIN047MH9.
[4] Research supported by EU FET Project ARRIVAL FP6-021235-2 and by the Dutch BSIK-BRICKS project.

1 Introduction

Wireless networks are used in many areas of practical interest, such as mobile phone communication, ad-hoc networks, and radio broadcasting. Recent advances in miniaturization of computing devices equipped with short range radios have raised strong interest in sensor networks for their practical relevance (environment control, accident monitoring etc.) [1, 8, 10]. One of the main issues concerning wireless networks is data gathering, i.e. collecting data from multiple nodes in a central sink node, which may process the data or act as gateway to other networks [6]. In the Wireless Gathering Problem (WGP) that we consider here, a static wireless network with several stations (*nodes*) and one *sink* is given. The network is modeled by an undirected, unweighted graph, where it is assumed that the distances in the graph approximate well the distances in the physical network. Through radio signals, stations can communicate to nodes within *transmission radius* d_T, the distance over which the signal is strong enough to send data. A typical difference with wired networks is that in wireless networks signals can interfere. The *interference radius* d_I ($\geq d_T$) is the distance over which the radio signal is strong enough to interfere with other radio signals. Interference causes data loss which results in lower throughput, increased packet delays and higher energy consumption [10].

Over time data packets arrive at nodes which have to be gathered at the base station. The problem consists of constructing a schedule without interference which determines for each packet both route and times at which it is sent, such as to minimize the maximum completion time of a packet.

Following earlier papers in data gathering [2, 3] we assume that stations have a common clock, hence time can be divided into rounds. Also, each node cannot send and receive during the same round(i.e. it is equipped with a half-duplex interface). Typically, not all nodes are within transmission radius from each other and the sink, so that packets have to be sent through several nodes before being gathered at the sink; this is called *multi-hop* routing. We assume that the stations are provided with the necessary information on the graph structure, such as path distances. Such information is available in a centralized setting, but can also be obtained in a distributed setting [9].

WGP was introduced by Bermond et al. [3] in the context of wireless access to the Internet in villages. The authors proved that minimizing the completion time is **NP**-hard and presented an algorithm with *asymptotic* approximation ratio at most 4, or 3 if $d_I = d_T$. The authors do not consider the case in which packets are released over time. Bermond et al. [4, 5] considered the so-called uniform-WGP, which has the extra assumptions of a single packet at each node and no release times, for which an optimal algorithm is presented in Bermond et al. [4] if the graph is a chain with the sink at one end. In case the sink is in the middle of the chain they give near optimal algorithms. Bar-Yehuda et al. [2] considered distributed algorithms for WGP. Their model is a special case of our model, where $d_I = 1$ and there are no release dates. Finally, Kumar et al. [7] give distributed approximation algorithms for wireless gathering in a different model in which the point-to-point communication is assumed to be symmetric.

We present an on-line polynomial-time greedy algorithm which for arbitrary release times gives a 4-approximation in general and a 3-approximation when $d_I = d_T$. Our results improve over those of Bermond et al. [3] only slightly when all packets are released at time 0. However, our algorithm is simple and the approximation ratios hold for arbitrary release times. Both our algorithm and that of [3] send all packets along a shortest path to the sink. We prove that no shortest path following algorithm can have approximation ratio better than 4. Thus, within this class our algorithm is best possible. Furthermore, we prove that our algorithm is optimal on a chain when $d_T = 1$ and the sink is at one end of the chain. All these results are found in Section 3. The complexity of all other variations of the problem on a chain and on a tree is to the best of our knowledge still open. These and some other challenging research opportunities on WGP conclude the paper in Section 4. We start by formalizing the problem.

2 Mathematical formulation

In WGP we are given a graph $G = (V, E)$ with $|V| = n$, a *sink* $s \in V$, and a set of packets $M = \{1, 2, \ldots, m\}$. Each $j \in M$ has an *origin* $v_j \in V$ and a *release time* $r_j \in \mathbb{Z}_+$ at which it enters the network. The *distance* $d(u, v)$ between nodes u and v is the length of a shortest path from u to v in G (the graph is unweighted). Also given are two positive integers d_T, d_I with $d_T \leq d_I$ that model the transmission and interference radius, respectively.

We assume that time is discrete; we call a time unit a *round*. The rounds are numbered $0, 1, \ldots$. During each round a node may either be *sending* a packet, be *receiving* a packet or be *inactive*. If $d(u, v) \leq d_T$ then u can send some packet j to v during a round. If node u sends packet j to v in some round, then the pair (u, v) is called a *call* of j during that round. Two calls (u, v) and (u', v') *interfere* if $d(u', v) \leq d_I$ or $d(u, v') \leq d_I$; otherwise they are *compatible*. We assume that packets cannot be aggregated. Notice that since there is a single destination node, there is no need to maintain more than one copy of a packet; in particular, without loss of generality a sender node can delete the packet after a successful transmission to a neighbor.

The solution of WGP is a schedule of compatible calls such that all packets arrive at the sink. Given a schedule, let v_j^t be the node of packet j at time t. The quantity $C_j := \min\{t : v_j^t = s\}$ is the *completion time* of packet j. The objective is minimizing $\max_j C_j$ (*makespan*).

In the off-line version all packet information is known at time 0, in the on-line version information about a packet becomes known only at its release time. The off-line WGP is equivalent to a one-to-many personalized broadcast problem: a time reverse gathering schedule provides a one-to-many personalized broadcast schedule. We introduce some extra notation. Let $\delta_j := \left\lceil \frac{d(v_j,s)}{d_T} \right\rceil$, the minimum number of calls required for packet j to reach s. The *critical radius* R^* is the greatest integer R such that no two nodes at distance at most R from s can receive a message in the same round. It is not hard to see that $R^* \geq \left\lfloor \frac{d_I - d_T}{2} \right\rfloor$ (see e.g. [3]). The *critical region* is the set $\{v \in V \mid d(s,v) \leq R^*\}$. Finally we define $\gamma := 1 + \left\lceil \frac{d_I+1}{d_T} \right\rceil$, and $\gamma^* := \left\lceil \frac{R^*+1}{d_T} \right\rceil \geq 1$. Roughly stated, γ gives a bound on the number of rounds during which a packet needs to be forwarded before a new packet can be safely forwarded from the same origin. On the other hand, γ^* gives the number of rounds during which each packet has to move inside the critical region. These parameters will play an important role in our analysis.

3 A greedy algorithm

We present a greedy algorithm which assigns packets to calls according to some priority ordering. We specify the ordering later, since our first results hold for any priority ordering.

Algorithm 1 (Priority Greedy (PG)) *In every round, consider the available messages in order of decreasing priority, and send each message as far as possible along a (possibly prefixed) shortest path from its current node to s, without creating interference with any higher-priority message.*

We analyze the worst-case approximation ratio of PG. Packet j is said to be *blocked* in round t if, in round t, j is not sent over distance d_T, or if j is not sent to s if $d(v_j^{t-1}, s) \leq d_T$. We define the following *blocking relation* on a PG schedule: $k \prec j$ if in the last round in which j is blocked by the transmission of higher order packets in that round, k is amongst these packets the one closest to j (ties broken arbitrarily).

The blocking relation induces a directed graph $F = (M, A)$ on M with an arc (k, j) for each $k, j \in M$ such that $k \prec j$. Observe that for any PG schedule F is a directed forest and the root of each tree of F is a message which is never blocked. For each j let $T(j) \subseteq F$ be the tree of F containing j, $b(j) \in M$ the root of $T(j)$, and $P(j)$ the path in F from $b(j)$ to j. Let $h(j)$ be the length of

$P(j)$, $\pi_j = \min\{\delta_j, \gamma^*\}$, and $R_j = r_j + \delta_j - \pi_j$. We derive an upper bound on the completion time C_j in a PG schedule.

Lemma 1 *For each packet $j \in M$, $C_j \le R_{b(j)} + \sum_{i \in P(j)} \min\{\delta_i, \gamma\}$.*

PROOF. The proof is by induction on $h(j)$. Any packet j with $h(j) = 0$ is never blocked, hence $b(j) = j$, and the lemma is obviously true. Otherwise, let t be the last round in which j is blocked by packet k, $k \prec j$. By definition of the blocking relation we have $d(v_j^t, v_k^t) \le d_T + d_I$. If $d(v_j^t, v_k^t) > d_I + 1$ then j, although blocked, is sent to v_j^{t+1} with $d(v_j^{t+1}, v_k^t) = d_I + 1$. Also, $d(v_k^t, s) \le (C_k - t)d_T$, otherwise k would not reach s by time C_k. From time $t+1$ on, j is forwarded to s over distance d_T each round, reaching s at time

$$
\begin{aligned}
C_j &\le t + 1 + \left\lceil \frac{d(v_k^t, s) + d(v_j^{t+1}, v_k^t)}{d_T} \right\rceil \le t + 1 + C_k - t + \left\lceil \frac{d_I + 1}{d_T} \right\rceil \\
&= C_k + 1 + \left\lceil \frac{d_I + 1}{d_T} \right\rceil = C_k + \gamma.
\end{aligned}
$$

Also, $C_j \le C_k + \delta_j$, since after k reaches s, j will need no more than δ_j rounds to reach s. Thus $C_j \le C_k + \min\{\delta_j, \gamma\}$ and the lemma follows by applying the induction hypothesis to C_k. □

Now we derive lower bounds on the optimal cost. Let C_j^* denote the completion time of packet j in an optimal solution.

Lemma 2 *For any $S \subseteq M$, $S \neq \emptyset$, there is $k \in S$ such that* $\max_{j \in S} C_j^* \ge R_k + \sum_{j \in S} \pi_j$.

PROOF. Since in every round at most one packet can move inside the critical region, any feasible solution to WGP gives a feasible solution to a preemptive single machine scheduling problem in which the release time of job j (corresponding to packet j) is R_j and its processing time is π_j. Ignoring interference outside the critical region can only decrease the optimum cost, thus a lower bound on the scheduling cost is also a lower bound on the gathering cost.

Let k be the first packet in S entering or being released in the critical region in the optimal schedule. In the scheduling relaxation, the makespan is at least the time at which the first job starts processing plus the sum of the processing times which is precisely what is stated in the lemma. □

Lemmas 1 and 2 hold for any priority ordering of the packets. Now we analyze the approximation ratio for PG with priority based on R_j: if $R_j < R_k$ then j will have higher priority than k (ties can be broken arbitrarily). We call this algorithm PG_R. Notice that PG_R is an on-line polynomial time algorithm.

Theorem 3 PG_R *is a γ/γ^*-approximation algorithm for WGP.*

PROOF. Let j be the packet having maximum C_j. Applying Lemma 2 with $S = T(j)$, the tree containing j in the blocking relation induced forest, yields

$$\max_{i \in T(j)} C_i^* \geq R_k + \sum_{i \in T(j)} \pi_i \tag{1}$$

where k is some packet in $T(j)$. On the other hand, by using Lemma 1,

$$C_j \leq R_{b(j)} + \sum_{i \in P(j)} \min\{\delta_i, \gamma\} \leq R_{b(j)} + \frac{\gamma}{\gamma^*} \sum_{i \in P(j)} \pi_i \leq R_k + \frac{\gamma}{\gamma^*} \sum_{i \in P(j)} \pi_i \tag{2}$$

since $R_{b(j)} \leq R_i \ \forall i \in T(j)$, $b(j)$ being the root of $T(j)$. The theorem follows by direct comparison of (1) and (2). $\square$

Corollary 4 PG_R *is* 4*-approximate for general WGP and* 3*-approximate if $d_I = d_T$.*

PROOF. We distinguish several cases:
Case 1: If $d_I \leq 2d_T - 1$ then $\gamma = 3$ while $\gamma^* \geq 1$, which in particular proves the 3-approximation in case $d_I = d_T$.
Case 2: $d_I \leq 3d_T - 1$. Then $\gamma \leq 4$ and $\gamma^* \geq 1$.
Case 3: $\ell d_T \leq d_I \leq (\ell+2)d_T - 1$ for any odd integer $\ell \geq 3$. Then $\gamma \leq \ell + 3$ and as $R^* \geq \lfloor \frac{d_I - d_T}{2} \rfloor$ we have $\gamma^* \geq (\ell+1)/2$, and $\gamma/\gamma^* \leq 2(\ell+3)/(\ell+1) \leq 3$. $\square$

The analysis shows that the ratio $\gamma/\gamma^* = 4$ only if $d_I/d_T \in [2, 3)$ and the ratio approaches 2 if d_I/d_T tends to infinity.

Corollary 5 PG_R *is optimal if G is a chain with s as an extreme and $d_T = 1$.*

PROOF. If G is a chain and s an extreme, then the critical radius is $d_I + 1$. Thus, for $d_T = 1$ we have $\gamma^* = d_I + 2$. The claim follows since $\gamma = d_I + 2$ for $d_T = 1$. $\square$

$\mathrm{PG_R}$ sends packets over shortest paths. We show that no algorithm which sends each packet j over a shortest path from v_j to s can be better than 3-approximate if $d_I = d_T$ and 4-approximate if $d_I > d_T$. This means that to find algorithms with lower approximation ratios than $\mathrm{PG_R}$, packets need to be diverged from their shortest path to the sink if this path becomes congested.

First, consider the example of Figure 1 with $d_I = d_T = 1$. Nodes u_1, u_2, u_3 have $m/3$ packets each. Any shortest paths following algorithm sends all packets via u, yielding $\max_j C_j = 3m$. On the other hand a solution with no packet passing u has $\max_j C_j^* \leq 3 + m$. The example can easily be extended for arbitrary $d_I = d_T$.

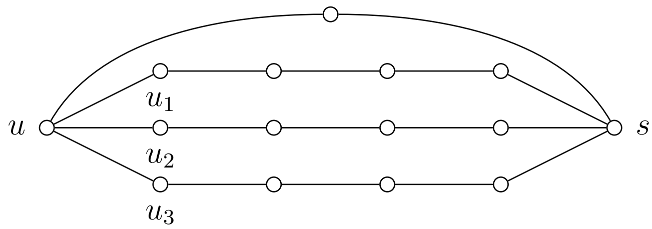

Figure 1. $d_I = d_T = 1$.

In case $d_I > d_T$ consider Figure 2 with $d_I = 2$ and $d_T = 1$. The nodes $u_1, \ldots, u_m$ each have 1 packet. Any shortest paths following algorithm sends all packets via u, yielding $\max_j C_j = 4m$. A solution with no packet passing u has $\max_j C_j^* \leq 4 + m$. The example can easily be extended for arbitrary $d_I = 2d_T$.

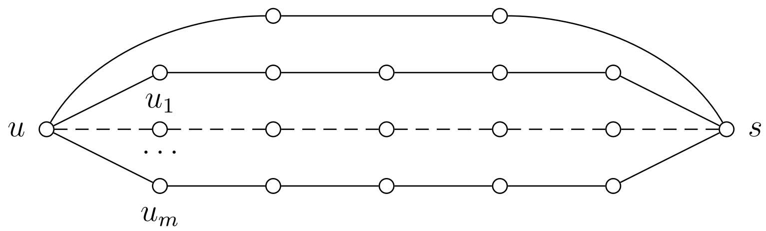

Figure 2. $d_I = 2, d_T = 1$.

In these examples the optimal schedule sends each packet over a path of length exceeding the length of its shortest path by at most 1. This may suggest to consider algorithms which send packets over paths whose length does not exceed their shortest path length by some constant k. However, as can easily be verified, for each constant k we could change the length of the paths in the examples above, such that the optimal schedule sends each packet over a path whose length exceeds the shortest path length by $k + 1$.

Improvement on the approximation ratio should come from algorithms that avoid congested paths. One such idea is to use not only the shortest path but the k shortest paths, whichever of them is least congested. However, again it is not difficult to adapt Figure 2, to show that in case $d_I > d_T$ choosing any of k shortest paths, for fixed k, leaves the lower bound of 4 on the ratio unchanged.

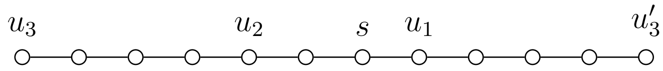

Figure 3. PG is not optimal on a chain if the sink is not at an extreme.

$\mathrm{PG_R}$ can be non-optimal on a chain if the sink is not at an extreme node of the chain. Consider the instance given by the graph in Figure 3. Let $d_T = 1$ and $d_I = 2$ and assume packets are released in u_1, u_2 and u_3 at time 0. $\mathrm{PG_R}$ would first send the packet in u_1 to the sink, and then send the packet in u_2 to the sink, resulting in a makespan of 7, while in an optimal solution the packets in u_2 and u_3 are forwarded until the packet of u_2 reaches the sink at time 2, then the packets of u_3 and u_1 are forwarded simultaneously, yielding a cost of 6. Note that if the third packet would have been released at node u'_3 instead of at node u_3 then $\mathrm{PG_R}$ would have been optimal. This shows that any optimal algorithm for the problem on a chain should take into account the position of the other packets when deciding which of the two packets nearest to the sink to send first. The complexity of the problem on a chain remains open.

4 Open Problems

In this paper we designed a greedy algorithm $\mathrm{PG_R}$ for data gathering in wireless networks and proved it to be 4-approximate when minimizing maximum completion time. We also showed that $\mathrm{PG_R}$ has a best possible approximation ratio within the class of algorithms in which each packet is sent over a shortest path to the sink. It is a beautiful challenge to design algorithms that avoid congested paths, which have approximation ratios strictly less than 4 (or 3 if $d_I = d_T$). Our examples at the end of the previous section show that this is not so trivial. Observing the proof of Corollary 4, one could concentrate on the subclass of problems with $d_I/d_T \in [2, 3)$ to improve on the ratio of 4, since in all other cases $\mathrm{PG_R}$ has ratio at most 3.

In fact, $\mathrm{PG_R}$ is an on-line algorithm, and thus gives a *competitive ratio* of 4. However, there is a significant gap between this upper bound and a simple lower bound on the competitive ratio of any deterministic algorithm: we have constructed rather simple examples, that we omit here, which give lower bounds of 7/5 for $d_I > d_T$ and 4/3 for $d_I = d_T$.

All our results apply to general graphs; we have not considered specific graphs in depth. Specifically, the complexity of WGP on chains or trees is open, apart from the restricted case in Corollary 5. We believe that these problems are easy as well, but $\mathrm{PG_R}$ is not optimal on a chain for $d_T = 1$ and $d_I = 2$ as we have shown at the end of the previous section. The example there can easily be extended to any combination of d_I and d_T, except $d_I = d_T$, for which $\mathrm{PG_R}$ might in fact be optimal.

$\mathrm{PG_R}$ can be implemented using only local information, i.e. each node requires only information of nodes within interference radius. It is a challenge to study truly distributed algorithms, which use only information available at a node. As interference can not always be detected a priori in this model, algorithms should be able to accommodate for retransmissions of lost data. Bar-Yehuda et al. [2] designed such distributed randomized algorithms for WGP with $d_T = d_I = 1$ without release times (thus in an off-line setting). They derive bounds on the expected number of rounds required to gather all packets at the sink. It would be interesting to exploit ideas in this paper to design (randomized) distributed on-line algorithms with satisfactory competitive ratios.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[2] R. Bar-Yehuda, A. Israeli, and A. Itai. Multiple communication in multi-hop radio networks. *SIAM Journal on Computing*, 22(4):875–887, 1993.

[3] J. Bermond, J. Galtier, R. Klasing, N. Morales, and S. Pérennes. Hardness and approximation of gathering in static radio networks. *Parallel Processing Letters*, 16(2):165–183, 2006.

[4] J.-C. Bermond, R. C. Corrêa, and M.-L. Yu. Gathering algorithms on paths under interference constraints. In *Proc. 6th Italian Conf. on Algorithms and Complexity*, pages 115–126, 2006.

[5] J.-C. Bermond, J. Galtier, R. Klasing, N. Morales, and S. Pérennes. Gathering in specific radio networks. In *Huitièmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, pages 85–88, Trégastel, France, 2006.

[6] M. Haenggi. Opportunities and challenges in wireless sensor networks. In *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. CRC Press, Boca Raton, 2004.

[7] V. S. Anil Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. End-to-end packet-scheduling in wireless ad-hoc networks. In J. I. Munro, editor, *Proc. 15th Symp. on Discrete Algorithms*, pages 1021–1030, 2004.

[8] K. Pahlavan and A. H. Levesque. *Wireless information networks*. Wiley, New York, 1995.

[9] C. E. Perkins. *Ad hoc networking*. Addison-Wesley, Boston, 2001.

[10] A. S. Tanenbaum. *Computer Networks*. Prentice Hall, 4th edition, 2003.