



**HAL**  
open science

## Fast Data Gathering in Radio Grid Networks

Jean-Claude Bermond, Nicolas Nisse, Patricio Reyes, Hervé Rivano

► **To cite this version:**

Jean-Claude Bermond, Nicolas Nisse, Patricio Reyes, Hervé Rivano. Fast Data Gathering in Radio Grid Networks. [Research Report] 2009. inria-00363908v1

**HAL Id: inria-00363908**

**<https://inria.hal.science/inria-00363908v1>**

Submitted on 24 Feb 2009 (v1), last revised 26 Mar 2009 (v4)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

## *Fast Data Gathering in Radio Grid Networks*

J-C. Bermond — N. Nisse — P. Reyes — H. Rivano

**N° 6851**

February 2009

Thème COM



*R*  
apport  
de recherche



## Fast Data Gathering in Radio Grid Networks

J-C. Bermond, N. Nisse, P. Reyes\* , H. Rivano

Thème COM — Systèmes communicants  
Équipe-Projet Mascotte

Rapport de recherche n° 6851 — February 2009 — 14 pages

**Abstract:** The aim of this paper is to design efficient gathering algorithms (data collection) in a Base Station of a wireless multi hop grid network when interferences constraints are present. We suppose the time is slotted and that during one time slot (step or round) each node can transmit to one of its neighbor at most one data item. Each device is equipped with an half duplex interface; so a node cannot both receive and transmit during a step. Each node is also equipped with unidirectional antennas which create interferences. During a step (round) only non interfering transmissions can be done. We mainly consider the primary node model where a node interferes with its neighbors, the non interfering calls done during a step will form a matching. The aim is to minimize the minimum completion time (makespan or number of steps need to send all messages to the base station. The best known algorithm for grids was a multiplicative 1.5-approximation algorithm. We give a very simple +2 approximation algorithm and then a more involved +1 approximation algorithm.

**Key-words:** Sensor Networks, gathering, makespan, grid.

This work has been partially funded by European project IST/FET AEOLUS.

\* Funded by Conicyt(Chile)/INRIA grant.

**Résumé :** Nous présentons des algorithmes efficaces pour la collecte d'informations par une station de base au sein d'un réseau sans-fil multi sauts en présence d'interférences. Nous nous focalisons sur les réseaux en grille car ils sont un bon modèle des réseaux d'accès comme des réseaux aléatoires de capteurs. Le temps est divisé en étapes élémentaires. Au cours d'une étape, un nœud peut transmettre au plus un message à l'un de ces voisins. Chaque appareil est équipé d'une interface half duplex et ne peut donc émettre et recevoir à la même étape. Ainsi, au cours d'une étape, l'ensemble des transmissions valides induit un couplage de la grille. Le problème consiste à minimiser le nombre d'étapes nécessaires à la collecte de tous les messages par la station de base. Le meilleur algorithme connu était une  $3/2$  approximation. Nous donnons un algorithme très simple qui approche l'optimum à 2 près, puis nous présentons un algorithme plus évolué qui est une  $+1$  approximation. Nos résultats sont valides lorsque les appareils ne disposent d'aucune mémoire tampon et doivent retransmettre un message à l'étape suivant sa réception.

**Mots-clés :** Sensor Networks, gathering, makespan, grid.

## 1 Introduction

We address here the challenging problem of gathering information in a Base Station (denoted BS) of a wireless multi hop grid network when interferences constraints are present. This problem is also known as data collection and is particularly important in sensor networks but also in access networks. The communication network is modeled by a graph. Here we consider grid topologies as they model well bot access networks and also random networks (which approximatively behave like if the nodes were on a grid see ([KLNP05])). We suppose the time is slotted and that during one time slot (step or round) each node can transmit to one of its neighbor at most one data item (referred in what follows as a message). In our model we allow each vertex of the grid to have any number of messages to transmit : zero if it is not concerned (sleeping station or no sensor at this node or failed device) one or many. We also suppose that each device (sensor, stations,...) is equipped with an half duplex interface; so a node cannot both receive and transmit during a step. Each node is also equipped with unidirectional antennas which create interferences along the transmission direction at some distance  $d_I$ . During a step (round) only non interfering transmissions can be done. For example in the primary node model where a node interferes with its neighbors, the non interfering calls done during a step will form a matching (set of independent edges). Our aim is to design algorithms to do a gathering under such hypotheses, which minimize the makespan or completion time (that is the number of steps in order the base station has received all the messages).

## 2 Related Work

The problem was introduced with different variants in many papers (see the survey [BKK<sup>+</sup>09]). Our article follows the work and hypotheses of [RS07] where a multiplicative 1.5-approximation algorithm is presented for grids (with primary interference node model), which was the best currently known result. In [RS07, RS08] the problem is also solved for path and star network topologies. In [FFM04], the same model is considered at least in their results. They present optimal (polynomial) algorithms for path and tree networks and a 2-approximation for general networks. Furthermore, they do not allow in the algorithms that they present, buffering in intermediate nodes. Their work has been extended to general graphs in [GR06] where an optimal algorithm is provided also without buffering, but in the uniform case where each node have one message for being sent to the BS.

The case of omnidirectional antennas has been extensively studied (see [BGK<sup>+</sup>06]) under the following assumptions: one message per node, authorized message buffering in intermediate nodes and different transmission and interference ranges. Lower bounds on the time to gather are given and NP-hardness is proved in [BGK<sup>+</sup>06] for general graphs. Moreover, a 4-approximation algorithm for this case is presented. The on line version is considered in [BKMS08] where a 4-approximation algorithm is given. For grid networks, the case is solved in [BP05] the base station being at the center of the grid. Gathering in grids is also considered within a continuous model in [GPRR08]. The authors fol-

lows the same assumptions of [BGK<sup>+</sup>06], but in this case, the messages are considered as flow units to gather.

On the other hand, in [BHW00, MPS95], the authors deal with a similar problem in grid networks when no buffering is considered (known as hot-potato routing). However, each node can successfully receive and transmit more than one message simultaneously.

### 3 Our results

Here we considerably improve the result on the grid; we present the results for the primary node model but they can be easily extended to any distance of interference. We first give a very simple +2 approximation algorithm and then a more involved +1 approximation algorithm. Furthermore our algorithm has no buffering and a message arrives at most one step after what will happen if we have no interferences. So the average time is also very good.

One helpful idea is to actually study the related one-to-all personalized broadcast problem in which the BS wants to communicate different data items to each other node in the network. Solving the above dissemination problem is equivalent to solve data gathering in sensor networks. Indeed, let  $T$  denote the makespan (delay), that is, the largest step used by a personalized broadcast algorithm; a gathering schedule with delay  $T$  consists in scheduling a transmission from node  $y$  to  $x$  during slot  $t$  iff the broadcasting algorithm schedules a transmission from node  $x$  to  $y$  during slot  $T - t + 1$ , for any  $t$  with  $1 \leq t \leq T$ .

## 4 Preliminaries

### 4.1 Notations

In the following, we consider a  $n \times n$  grid  $G = (V, E)$  where vertices are given their natural coordinates. More precisely, the base station  $BS$  has coordinates  $(0, 0)$ , and any vertex  $v$  has coordinates  $(x_v, y_v)$ . A vertex  $v$  is *above* (resp., *below*)  $w \in V$  if  $y_v \geq y_w$  (resp., if  $y_v \leq y_w$ ). Similarly,  $v$  is *to the right* (resp., *to the left*) of  $w \in V$  if  $x_v \geq x_w$  (resp., if  $x_v \leq x_w$ ). Finally, a vertex  $v$  is *nearer to the source* than  $w \in V$  is  $d(v, BS) \leq d(w, BS)$ , where  $d(u, v)$  denotes the classical distance between nodes  $u$  and  $v$ .

We consider a set of  $M \geq 0$  messages  $\mathcal{M}$  that must be sent from the source  $BS$  to some destination nodes. Let  $dest(m) \in V$  denote the destination of  $m \in \mathcal{M}$ . We use  $d(m)$  to denote  $d(dest(m), BS)$ . A message  $m \in \mathcal{M}$  is *lower* (resp., *higher*) than  $m' \in \mathcal{M}$  if  $dest(m)$  is below (resp., above)  $dest(m')$ .  $m$  is *righter* (resp., *leftier*) than  $m'$ , if  $dest(m)$  is to the right (resp., to the left) of  $dest(m')$ .  $m \preceq m'$  if  $dest(m)$  is nearer to the source than  $dest(m')$ . We suppose in what follows that the messages are ordered by non increasing distance of their destination nodes, and we note  $\mathcal{M} = \{m_1, \dots, m_M\}$  where  $m_j \preceq m_i$  for any  $i \leq j \leq M$ , and  $v_i \in V$  denotes the destination node of message  $m_i$ .

$S \odot S'$  denotes the sequence obtained by concatenation of two sequences  $S$  and  $S'$ .

## 4.2 Lower bound

Consider a model without interferences, but where the source can only send one message per step. Whatever be the broadcasting scheme, a message  $m$  sent at step  $t \geq 1$  will be received at step  $t' \geq d(m) + t - 1$ . A broadcasting scheme is said *greedy* if, given an ordering  $\mathcal{S}$  of the messages, the source sends one message per step, in the ordering  $\mathcal{S}$ , and each message follows a shortest path toward its destination node. Note that, in the model without interferences, if the messages follow shortest paths, a vertex will never receive more than one message per step.

**Definition 1**  $LB = \max_{i \leq M} d(m_i) + i - 1$ .

**Lemma 1** *In the model without interferences, when the source emits at most one message per step, a greedy algorithm following the ordered sequence of messages  $(m_1, m_2, \dots, m_M)$  is optimal, with makespan  $LB$ .*

**Proof.** Clearly, sending the messages in the ordering of the sequence  $(m_1, m_2, \dots, m_M)$  along shortest paths will achieve such a makespan. Now, let us consider an optimal schedule of the messages  $(s_1^*, \dots, s_M^*)$  different from  $(m_1, m_2, \dots, m_M)$  and let  $i \geq 1$  be the smallest integer such that  $s_i^* \neq m_i = s_j^*$  ( $j > i$ ). We prove that sending the messages in the ordering of the sequence  $(s_1^*, \dots, s_{i-1}^*, s_j^*, s_{i+1}^*, \dots, s_{j-1}^*, s_i^*, s_{j+1}^*, \dots, s_m^*)$  does not increase the makespan. Indeed, only the  $i^{\text{th}}$  and  $j^{\text{th}}$  messages differ and  $\max\{d(s_j^*) + i - 1, d(s_i^*) + j - 1\} \leq d(s_j^*) + j - 1$  because  $d(v_i, BS) = d(s_j^*) \geq d(s_i^*)$  and  $j > i$ . By iterating this process, we get that the ordering of the sequence  $(m_1, m_2, \dots, m_M)$  is also optimal. ■

**Corollary 1** *In the model with interferences, no algorithm can achieve a makespan less than  $LB$ .*

## 5 Algorithms

### 5.1 Horizontal-Vertical broadcasting

From now on, we consider the model with interferences at distance  $d_I = 1$ . First, we present a very simple broadcasting scheme that we prove to be sufficient to obtain a good approximation of the optimal makespan.

Given a message whose destination node  $v$  has coordinates  $(x, y)$ , the message is sent *horizontally* to  $v$  if it follows the shortest path from  $BS$  to  $v$  passing through  $(x, 0)$ . The message is sent *vertically* if it follows the shortest path from  $BS$  to  $v$  passing through  $(0, y)$ .

**Definition 2** *A Horizontal-Vertical broadcasting scheme, or HV-scheme, takes an ordering  $\mathcal{S}$  of  $\mathcal{M}$  as an input and proceeds as follows. A direction, horizontal or vertical, is chosen for the first message. Then, the source sends one message every  $d_I$  steps in the ordering  $\mathcal{S}$  and alternating horizontal and vertical messages.*

Let us do some easy remarks about any HV-scheme. Consider two distinct messages sent by the source  $x$  time-slots apart. Since these messages follow shortest paths, while the first message has not reached its destination, both messages are separated by a distance at least  $x$ . Hence,



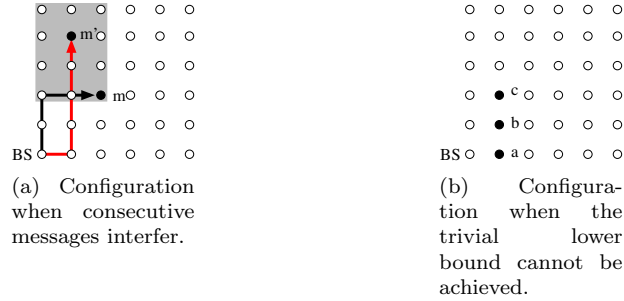


Figure 1: Two particular configurations

**Claim 1** *In a HV-scheme, only consecutive messages may interfere.*

Now, let us characterize forbidden and acceptable configurations in HV-scheme. Assume that two messages are sent consecutively. It is possible to guess the respective positions of their destination nodes by knowing whether both messages interfere or not. In Figure 1(a), nodes in the grey part are the nodes that are higher and left than  $y$ . Figure 1(a) illustrates the following Claim.

**Claim 2** *Let  $m, m'$  be two messages sent consecutively by a HV-scheme, with  $m$  sent vertically and  $m'$  sent horizontally.  $m$  and  $m'$  interfere if and only if their destinations are distinct and  $m'$  is higher and left than  $m$ .*

Before continuing, let us remark that there exist configurations for which no gathering protocol can achieve better makespan than  $LB + 1$ . Figure 1(b) represents such a configuration. Indeed, in Figure 1(b), the three destinations  $a, b$  and  $c$  have coordinates  $(1, 0), (1, 1)$  and  $(1, 2)$ , and  $LB = 3$ . However, to achieve such a makespan, the first message must be sent to  $c$  (because  $c$  is at distance 3 from  $g$ ) and the second message must be sent to  $b$  (because the message start after the first step and must go at distance 2). To avoid collision, the only possibility is to send the first message vertically, and the second one horizontally. Hence, the last message cannot reach  $a$  before step 4.

## 5.2 +2 approximation

Recall that  $(m_1, \dots, m_M)$  denotes the ordering of the messages in the non increasing ordering of the distance to their destinations. In this section, we give the Algorithm *TwoApprox*, depicted in Figure 2, that computes an ordered sequence  $\mathcal{S} = (s_1, \dots, s_m)$  of the messages such that

- HV-scheme( $\mathcal{S}$ ), sending the last message vertically, broadcasts the messages without collisions, and
- $m_i \in \{s_{i-2}, s_{i-1}, s_i, s_{i+1}, s_{i+2}\}$  for any  $i \leq M$ , and  $s_M \in \{m_{M-1}, m_M\}$

These two properties imply the following theorem.

**Theorem 1** *There is an ordering  $\mathcal{S}$  of the messages, such that HV-scheme( $\mathcal{S}$ ) achieves  $LB + 2$ .*

**Input:**  $\mathcal{M} = \{m_1, \dots, m_M\}$ , the set of messages ordered in non increasing distance order  
**Output:**  $(s_1, \dots, s_M)$  an ordered sequence of  $\mathcal{M}$  such that  $m_i \in \{s_{i-2}, s_{i-1}, s_i, s_{i+1}, s_{i+2}\}$  for any  $i \leq M$   
**begin**  
  **Case**  $M = 0$  **return**  $\emptyset$   
  **Case**  $M = 1$  **return**  $(m_1)$   
  **Case**  $M \geq 2$   
    Let  $\mathcal{O} \odot p = \text{TwoApprox}(\{m_1, \dots, m_{M-2}\})$   
    Let  $q$  be the lowest message in  $\{m_{M-1}, m_M\}$  and let  $r$  be the other one  
    **if**  $p$  is higher than  $q$  **return**  $\mathcal{O} \odot (p, q, r)$   
    **else return**  $\mathcal{O} \odot (m_{M-1}, p, m_M)$   
**end.**

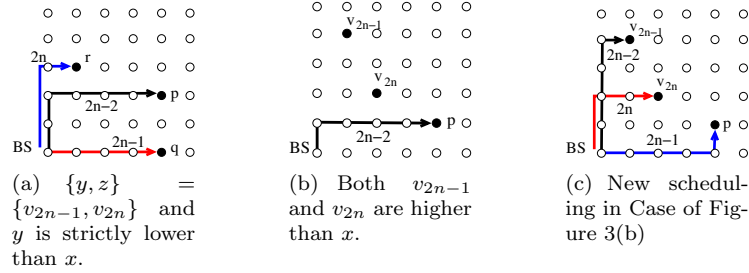
Figure 2: Algorithm *TwoApprox*

Figure 3:  $2n - 2$  messages have been scheduled, finishing with the one to  $x \in \{v_{2n-2}, v_{2n-3}\}$ . When the next two messages must be scheduled, two cases occur according to the position of  $v_{2n-1}$  and  $v_{2n}$  relatively to  $x$ . In the figures, an arrow with label  $i$  represents the route of the  $i^{\text{th}}$  message.

**Proof.** To prove the correctness of Algorithm *TwoApprox*, we proceed by induction on  $M$ . If  $M \leq 2$ , the result holds obviously. Let us assume that the ordering of the sequence computed by  $\text{TwoApprox}(\{m_1, \dots, m_{M-2}\})$  satisfies the two properties. Let  $p$  be the last message of this sequence. By the induction hypothesis,  $p \in \{m_{M-3}, m_{M-2}\}$  is sent vertically. Let  $t$  be the next to last message in this sequence, if any. By Claim 2,  $p$  must be higher or left than  $t$ . The sequence is denoted by  $\mathcal{O} \odot p = \mathcal{O}' \odot (t, p)$ .

Let  $q$  be the lowest message in  $\{m_{M-1}, m_M\}$  and let  $r$  be the other one. If  $p$  is higher than  $q$ , it is sufficient to send  $q$  horizontally at step  $M - 1$ , and  $r$  vertically at step  $M$ . This case is depicted in Figure 3(a). Indeed, by Claim 1 only  $p$  and  $q$ , or  $q$  and  $r$  may interfere. By Claim 2, there are no interferences. It is easy to check that  $\mathcal{O} \odot (p, q, r)$  satisfies the properties. Otherwise,  $q$  and  $r$  are higher than  $p$ . Moreover, since  $q, r \preceq p$ ,  $q, r$  are higher and left than  $p$ . This case is depicted in Figure 3(b). In this case, instead of sending  $p$  at step  $M - 2$ , the source sends  $m_{M-1}$  vertically at step  $M - 2$ , then  $p$  horizontally  $M - 1$ , and then  $m_M$  vertically at step  $M$ . The transformation is depicted in Figure 3(c). Clearly,  $\mathcal{O} \odot (m_{M-1}, p, m_M)$  satisfies the properties. We prove that the corresponding HV-broadcasting performs without interferences. By Claim 1

only  $t$  and  $m_{M-1}$ , or  $m_{M-1}$  and  $p$ , or  $p$  and  $m_M$  may interfere. Since  $m_{M-1}$  is higher and letter than  $p$  that is higher or letter than  $t$ , by Claim 2,  $m_{M-1}$  interferes neither with  $t$  nor with  $p$ . Similarly,  $m_M$  is higher and letter than  $p$  and these messages do not interfere. ■

### 5.3 +1 approximation

In this section, we give the Algorithm *OneApprox*, depicted in Figure 5, that computes an ordered sequence  $\mathcal{S} = (s_1, \dots, s_m)$  of the messages such that

- HV-scheme( $\mathcal{S}$ ), sending the last message vertically, broadcasts the messages without collisions, and
- $m_i \in \{s_{i-1}, s_i, s_{i+1}\}$  for any  $i \leq M$  (in particular, either  $m_M = s_M$ , or  $m_{M-1} = s_M$  and  $m_M = s_{M-1}$ ).

**Theorem 2** *There is an ordering  $\mathcal{S}$  of the messages, such that HV-scheme( $\mathcal{S}$ ) achieves  $LB + 1$ .*

**Proof.** We prove that Algorithm *OneApprox* computes an ordered sequence of messages satisfying the two properties. We proceed by induction on  $M$ . An ordered sequence  $\mathcal{S} = (s_1, \dots, s_M)$  of  $\mathcal{M}$  satisfying the desired properties is said *valid*.  $\mathcal{S}$  is said  *$i$ -almost valid* if it satisfies all desired properties, but  $s_i$  interferes with  $s_{i+1}$ , for a unique  $i \leq M$ . If  $M \leq 2$ , the result holds obviously. Let us assume that the sequence *OneApprox*( $\{m_1, \dots, m_{M-2}\}$ ) satisfies the two properties. Let  $p$  be the last message of this sequence. By the induction hypothesis,  $p \in \{m_{M-3}, m_{M-2}\}$  is sent vertically. The sequence is denoted by  $(s_1, \dots, s_{M-2})$ .

For any  $i < M - 2$ ,  $\mathcal{O}_i$  denotes  $(s_1, \dots, s_{M-2-i})$ , i.e.,  $(s_1, \dots, s_{M-2}) = \mathcal{O}_i \odot (s_{i+1}, \dots, s_{2n-2})$ . By the induction hypothesis,  $s_{M-2} \in \{v_{2n-3}, v_{2n-2}\}$  and has been sent vertically. Let  $q$  be the lowest message in  $\{m_{M-1}, m_M\}$  and let  $r$  be the other one. We consider the 3 cases of Algorithm *OneApprox* (when  $M \geq 2$ ).

- a) If  $s_{M-2}$  is higher than  $q$ , it is sufficient to send  $q$  horizontally at step  $M-1$ , and  $r$  vertically at step  $M$ . This case is depicted in Figure 3(a). Indeed, by Claim 1 only  $s_{M-2}$  and  $q$ , or  $q$  and  $r$  may interfere. By Claim 2, there are no interferences. It is easy to check that  $\mathcal{O}_1 \odot (s_{M-2}, q, r)$  is valid.
- b) If  $q$  and  $r$  are higher than  $s_{M-2}$ , and  $s_{M-2} = m_{M-2}$ . Moreover, since  $q, r \preceq s_{M-2}$ ,  $q, r$  are higher and letter than  $s_{M-2}$ . This case is depicted in Figure 3(b). In this case, instead of sending  $s_{M-2}$  at step  $M-2$ , the source sends  $m_{M-1}$  vertically at step  $M-2$ , then  $s_{M-2}$  horizontally  $M-1$ , and then  $m_M$  vertically at step  $M$ . The transformation is depicted in Figure 3(c). Clearly,  $\mathcal{O}_1 \odot (m_{M-1}, s_{M-2}, m_M)$  is valid (in particular because  $s_{M-2} = m_{M-2}$ ). We prove that the corresponding HV-broadcasting performs without interferences. By Claim 1 only  $s_{M-3}$  and  $m_{M-1}$ , or  $m_{M-1}$  and  $s_{M-2}$ , or  $s_{M-2}$  and  $m_M$  may interfere. Since  $m_{M-1}$  is higher and letter than  $s_{M-2}$  that is higher or letter than  $s_{M-3}$ , by Claim 2,  $m_{M-1}$  interferes neither with  $s_{M-3}$  nor with  $s_{M-2}$ . Similarly,  $m_M$  is higher and letter than  $s_{M-2}$  and these messages do not interfere.

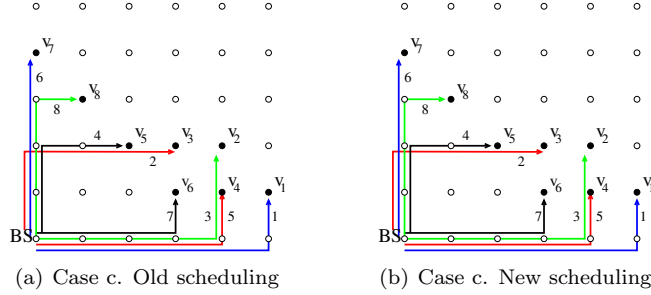


Figure 4: Case with recursive modifications of the scheduling

- c) If  $q$  and  $r$  are higher than  $s_{M-2}$ , and  $s_{M-2} \neq m_{M-2}$  (i.e.,  $s_{M-3} = m_{M-2}$  and  $s_{M-2} = m_{M-3}$ ). Moreover,  $s_{M-2}$  is sent vertically. This case is depicted in Figure 4 (with  $M = 8$ ).

By Claim 2 and because  $s_{M-3} \preceq s_{M-2}$ ,  $s_{M-3}$  is lower than  $s_{M-2}$ . Indeed, these messages do not interfere in the ordered sequence computed by Algorithm *OneApprox*. Then, it is possible to send messages  $m_{M-3}, m_{M-1}, m_{M-2}$  and  $m_M$ , alternatively horizontal and vertical (starting horizontally) without any interference between these four messages. Therefore, the scheduling  $\mathcal{O}_{M-4} \odot (m_{M-3}, m_{M-1}, m_{M-2}, m_M)$  is either valid or  $(M-4)$ -almost valid.

We prove that Subprocedure  $MakeValid(\mathcal{O}_{M-4} \odot (m_{M-3}, m_{M-1}, m_{M-2}, m_M), 2)$  computes an ordered sequence of  $\mathcal{M}$  that is valid. We need the following technical definition.

**Definition 3** Let  $i \in [2, \lfloor M/2 \rfloor]$ . An ordered sequence  $\mathcal{S} = (s_1, \dots, s_M)$  is  $i-1$ -good if it is  $(M-2i)$ -almost valid, and  $s_{M-2i+1} = m_{M-2i+1}$ ,  $s_{M-2i+2} = m_{M-2i+3}$  and  $s_M = m_M$ .

Let us do the following easy remarks.

1. In a  $i$ -good sequence  $\mathcal{S}$ , if  $s_{M-2i}$  does not interfere with  $s_{M-2i+1}$  then  $\mathcal{S}$  is valid.
2. In particular, if  $2i = M$ , a  $i-1$ -good sequence is valid.
3. Note that the sequence  $\mathcal{O}_{M-4} \odot (m_{M-3}, m_{M-1}, m_{M-2}, m_M)$  is 1-good.

Let  $i, 1 \leq i \leq \lfloor M/2 \rfloor$ . and let  $\mathcal{S}$  be a  $i-1$ -good sequence. By reverse induction on  $i$ , we prove that  $MakeValid(\mathcal{S}, i)$  will compute a valid sequence.

We recall that

$$\begin{aligned} \mathcal{S} &= (s_1, \dots, s_{M-2i-2}) \odot (s_{M-2i-1}, s_{M-2i}, s_{M-2i+1}, s_{M-2i+2}) \odot (s_{M-2i+3}, \dots, s_M) \\ &= (s_1, \dots, s_{M-2i-2}) \odot (s_{M-2i-1}, s_{M-2i}, m_{M-2i+1}, m_{M-2i+3}) \odot (s_{M-2i+3}, \dots, s_M) \end{aligned}$$

First, if  $2i = M$ , then  $s_{M-2i}$  does not interfere with  $s_{M-2i+1}$  (because  $s_{M-2i}$  is not defined).  $MakeValid(\mathcal{S}, i)$  returns  $\mathcal{S}$  which is valid by Remark 2.

If  $2i = M - 1$  and  $s_1$  and  $s_2$  do not interfere,  $MakeValid(\mathcal{S}, i)$  returns  $\mathcal{S}$  that is valid by Remark 2. Otherwise, if  $2i = M - 1$  and  $s_1$  and  $s_2$  interfere, then  $\mathcal{S} = (s_1, m_2, m_4) \odot (s_5, \dots, s_{2i+1})$ . This implies that  $s_1 = m_1$ . In this case,  $MakeValid(\mathcal{S}, i)$  returns  $(m_2, m_1, m_4) \odot (s_5, \dots, s_{2i+1})$ . Moreover, by parity and because  $s_{2i+1}$  is sent vertically,  $m_2$  is sent horizontally. By Claim 2 and because  $m_1 \preceq m_2 \preceq m_4$ ,  $m_1$  must be lower than  $m_2$  that is lower than  $m_4$ . By Claim 1, the only possible interferences may occur between  $m_1$  and  $m_2$ , or  $m_2$  and  $m_4$ . The respective positions of the destination nodes of  $m_1, m_2, m_4$  imply that this sequence is valid.

Now, assume  $2i < M - 1$ , and assume that, if  $\mathcal{S}$  is  $i$ -good,  $MakeValid(\mathcal{S}, i + 1)$  will compute a valid sequence. Assume that  $\mathcal{S}$  is  $i - 1$ -good. If  $s_{M-2i}$  and  $s_{M-2i+1}$  do not interfere, then  $MakeValid(\mathcal{S}, i)$  returns  $\mathcal{S}$  that is valid by Remark 2. Let us consider the case when  $s_{M-2i}$  and  $s_{M-2i+1}$  interfere. By parity,  $s_{M-2i}$  is sent vertically and  $s_{M-2i+1}$  horizontally. By Claim 2,  $s_{M-2i+1}$  is higher and lefter than  $s_{M-2i}$ . Moreover, since  $s_{M-2i+1}$  and  $s_{M-2i+2}$  do not interfere, then  $s_{M-2i+2}$  must be higher or lefter than  $s_{M-2i+1}$ . Similarly,  $s_{M-2i-1}$  must be either lower or righter than  $s_{M-2i}$ . There are two cases to be considered according to the value of  $s_{M-2i}$ . Recall that, because  $s_{M-2i+1} = m_{M-2i+1}$   $s_{M-2i} \in \{m_{M-2i}, m_{M-2i-1}\}$ .

c.1 If  $s_{M-2i} = m_{M-2i}$ .

Because of their respective positions, it is possible to send messages  $s_{M-2i-1}, s_{M-2i+1}, s_{M-2i}$  and then  $s_{M-2i+2}$ , alternatively horizontal and vertical (starting horizontally) without any interference between these 4 messages. In this case,  $MakeValid(\mathcal{S}, i)$  returns  $(s_1, \dots, s_{M-2i-2}) \odot (s_{M-2i-1}, s_{M-2i+1}, s_{M-2i}, s_{M-2i+2}) \odot (s_{M-2i+3}, \dots, s_M)$ . By previous remark, this sequence does not create any interference. Moreover, only  $s_{M-2i+1} = m_{M-2i+1}$  and  $s_{M-2i} = m_{M-2i}$  have been switched. Therefore, it is easy to check that it is valid.

c.2 If  $s_{M-2i} = m_{M-2i-1}$ . In this case, note that  $s_{M-2i-1} = m_{M-2i}$ . Because of their respective positions, it is possible to send messages,  $s_{M-2i}, s_{M-2i+1}, s_{M-2i-1}$  and then  $s_{M-2i+2}$  alternatively horizontal and vertical (starting horizontally) without any interference between these 4 messages. Hence, the only possible interference in the following sequence is between  $s_{M-2i-2}$  and to  $s_{M-2i}$ :  $\mathcal{S}' = (s_1, \dots, s_{M-2i-2}) \odot (s_{M-2i}, s_{M-2i+1}, s_{M-2i-1}, s_{M-2i+2}) \odot (s_{M-2i+3}, \dots, s_M)$ . It is easy to check that  $\mathcal{S}'$  is  $i$ -good. Besides,  $MakeValid(\mathcal{S}, i)$  returns  $MakeValid(\mathcal{S}', i + 1)$  that is valid by the induction assumption. ■

Now, we prove that Algorithm *OneApprox* performs in linear time, w.r.t., the number of messages.

**Theorem 3** *The time complexity of Algorithm OneApprox is  $O(M)$ .*

**Proof.** We note  $c(i)$  the time-complexity of  $OneApprox(\{m_1, \dots, m_i\})$ . We prove by induction on  $i$  that  $c(i) = O(i)$ . Let  $S = (s_1, \dots, s_M)$  be the ordered sequence computed by  $OneApprox(\mathcal{M})$ . The pivot  $s_{M-2P}$  of this sequence is the message such that  $s_{M-2P} = m_{M-2P}$ , and minimizing  $P$ . More

```

Input:  $\mathcal{M} = \{m_1, \dots, m_M\}$ , the set of messages ordered in non increasing distance order
Output:  $(s_1, \dots, s_M)$  an ordered sequence of  $\mathcal{M}$  such that  $m_i \in \{s_{i-1}, s_i, s_{i+1}\}$  for any  $i \leq M$ 
begin
  Case  $M = 0$  return  $\emptyset$ 
  Case  $M = 1$  return  $(m_1)$ 
  Case  $M \geq 2$ 
    Let  $\mathcal{O} \odot p = \text{OneApprox}(\{m_1, \dots, m_{M-2}\})$ 
    Let  $q$  be the lowest message in  $\{m_{M-1}, m_M\}$  and let  $r$  be the other one
    if  $p$  is higher than  $q$  return  $\mathcal{O} \odot (p, q, r)$ 
    else if  $p = m_{M-2}$  return  $\mathcal{O} \odot (m_{M-1}, p, m_M)$ 
    else
      /* This last case may occur only if  $M > 3$  */
      Let  $(s_1, \dots, s_{M-4}) \odot (m_{M-2}, m_{M-3}) = \text{OneApprox}(\{m_1, \dots, m_{M-2}\})$ 
      return  $\text{MakeValid}((s_1, \dots, s_{M-4}) \odot (m_{M-3}, m_{M-1}, m_{M-2}, m_M), 2)$ 
end.

```

Figure 5: Algorithm *OneApprox*

```

Input: An integer  $j$ ,  $1 < j \leq \lfloor M/2 \rfloor$ , and
  a  $j - 1$ -good sequence  $\mathcal{O} = (s_1, \dots, s_M)$  of a set of messages  $\{m_1, \dots, m_M\}$ ,
Output: A valid sequence of  $\mathcal{M}$ 
  if  $s_{M-2j}$  and  $s_{M-2j+1}$  do not interfere
    /* In particular, this case occurs if  $M - 2j = 0$  */
    return  $\mathcal{O}$ 
  else if  $s_{M-2j} = m_{M-2j}$ 
    /* In particular, this case occurs if  $M - 2j = 1$  */
    return  $(s_1, \dots, s_{M-2j-2}) \odot (s_{M-2j-1}, s_{M-2j+1}, s_{M-2j}, s_{M-2j+2}) \odot (s_{M-2j+3}, \dots, s_M)$ 
  else return
    /* This last case may occur only if  $M - 2j \geq 2$  */
    /* Note that, in this case,  $s_{M-2j} = m_{M-2j-1}$  and  $s_{M-2j-1} = m_{M-2j}$  */
     $\text{MakeValid}((s_1, \dots, s_{M-2j-2}) \odot (s_{M-2j}, s_{M-2j+1}, s_{M-2j-1}, s_{M-2j+2}) \odot (s_{M-2j+3}, \dots, s_M), j + 1)$ 
end.

```

Figure 6: MakeValid

precisely, we prove that  $c(M) = O(c(M - 2P) + O(2P))$ . If  $M \leq 1$ , the result is trivial. Let us assume  $M \geq 2$ . By induction, the computation of  $\text{OneApprox}(\{m_1, \dots, m_{M-2}\})$  takes time  $O(c(M - 2 - 2P') + O(2P'))$  where  $P'$  is the pivot of the obtained sequence. There are three possible cases corresponding to the three cases of the algorithm (when  $M \geq 2$ ).

- Clearly, in this case,  $c(M) = c(M - 2) + O(1)$ . Moreover, either  $P = P' + 1$  (if  $r = m_{M-1}$ ) or  $P = 0$  (if  $r = m_M$ ). In both cases, we get  $c(M) = O(c(M - 2P) + O(2P))$ .
- $c(M) = c(M - 2) + O(1)$  and  $P = P' + 1$ , thus  $c(M) = O(c(M - 2P) + O(2P))$ .

- c. Let  $c'(M)$  be the complexity of  $Makevalid((s_1, \dots, s_M), 2)$ . In this case,  $c(M) = c(M - 2) + c'(M)$  and  $P = 0$  (because in the computed sequence,  $s_M = m_M$ ). Finally, when executed  $Makevalid((s_1, \dots, s_M), 2)$ , the same subprocedure  $MakeValid$  is recursively executed until  $s_{M-2j}$  and  $s_{M-2j+1}$  do not interfere, or  $s_{M-2j} = m_{M-2j}$ . Therefore, it is executed at most  $P'$  times and each execution takes  $O(1)$ . Hence,  $c(M) = O(M - 2P) + O(2P) = O(M)$ .

■

## 6 Future works

In this paper, we have presented centralized algorithm for the minimum makespan personalized broadcasting in grid networks. In these settings, the problem is strictly equivalent to the data gathering problem. One can note that our network model assumes that an optimal MAC layer is available. It would be interesting to investigate on the behavior of the problems under weaker assumptions. Another direction to investigate is the online version of the problems. It is worth pointing out that, in this case, personalized broadcasting and gathering are no longer equivalent.

## References

- [BGK<sup>+</sup>06] J.-C. 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.
- [BHW00] Costas Busch, Maurice Herlihy, and Roger Wattenhofer. Hard-potato routing. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 278–285, Portland, Oregon, United States, 2000. ACM.
- [BKK<sup>+</sup>09] V. Bonifaci, R. Klasing, P. Korteweg, L. Stougie, and A. Marchetti-Spaccamela. *Graphs and Algorithms in Communication Networks*, chapter Data Gathering in Wireless Networks. Springer Monograph Springer-Verlag, 2009.
- [BKMS08] Vincenzo Bonifaci, Peter Korteweg, Alberto Marchetti-Spaccamela, and Leen Stougie. An approximation algorithm for the wireless gathering problem. *Operations Research Letters*, 36(5):605 – 608, 2008.
- [BP05] J.-C. Bermond and J. Peters. Efficient gathering in radio grids with interference. In *Septièmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'05)*, pages 103–106, Presqu'île de Giens, May 2005.
- [FFM04] C. Florens, M. Franceschetti, and R.J. McEliece. Lower bounds on data collection time in sensory networks. *Selected Areas in Communications, IEEE Journal on*, 22(6):1110–1120, 2004.

- 
- [GPRR08] C. Gomes, S. Pérennes, P. Reyes, and H. Rivano. Bandwidth allocation in radio grid networks. In *10èmes Rencontres Francophones sur les Aspects Algorithmiques de Télécommunications (AlgoTel'08)*, May 2008.
- [GR06] Luisa Gargano and Adele Rescigno. *Optimally Fast Data Gathering in Sensor Networks*, pages 399–411. 2006.
- [KLNP05] R. Klasing, Z. Lotker, A. Navarra, and S. Pérennes. From balls and bins to points and vertices. In *Proceedings of the 16th Annual International Symposium on Algorithms and Computation (ISAAC 2005)*, volume 3827 of *Lecture Notes in Computer Science*, pages 757–766. Springer Verlag, December 2005.
- [MPS95] Yishay Mansour and Boaz Patt-Shamir. Many-to-one packet routing on grids. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 258–267, Las Vegas, Nevada, United States, 1995. ACM.
- [RS07] Y. Revah and M. Segal. Improved algorithms for data-gathering time in sensor networks ii: Ring, tree and grid topologies. In *Networking and Services, 2007. ICNS. Third International Conference on*, page 46, 2007.
- [RS08] Yoram Revah and Michael Segal. Improved bounds for data-gathering time in sensor networks. *Computer Communications*, 31(17):4026–4034, November 2008.



## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Related Work</b>	<b>3</b>
<b>3</b>	<b>Our results</b>	<b>4</b>
<b>4</b>	<b>Preliminaries</b>	<b>4</b>
4.1	Notations . . . . .	4
4.2	Lower bound . . . . .	5
<b>5</b>	<b>Algorithms</b>	<b>5</b>
5.1	Horizontal-Vertical broadcasting . . . . .	5
5.2	+2 approximation . . . . .	6
5.3	+1 approximation . . . . .	8
<b>6</b>	<b>Future works</b>	<b>12</b>



---

Centre de recherche INRIA Sophia Antipolis – Méditerranée  
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex  
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier  
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq  
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique  
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex  
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex  
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex  
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex

---

Éditeur  
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-6399