

Optimal Time Data Gathering in Wireless Networks with Omni-Directional Antennas

Jean-Claude Bermond, Luisa Gargano, Stéphane Pérennes, Adele Rescigno,
Ugo Vaccaro

► **To cite this version:**

Jean-Claude Bermond, Luisa Gargano, Stéphane Pérennes, Adele Rescigno, Ugo Vaccaro. Optimal Time Data Gathering in Wireless Networks with Omni-Directional Antennas. SIROCCO2011, Jun 2011, Gdansk, Poland. Springer, 6796, pp.306-317, 2011, Lecture Notes in Computer Science - LNCS. <inria-00589687>

HAL Id: inria-00589687

<https://hal.inria.fr/inria-00589687>

Submitted on 30 Apr 2011

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.

Optimal Time Data Gathering in Wireless Networks with Omni-Directional Antennas

Jean-Claude Bermond^{1*}, Luisa Gargano², Stephane Perennes¹,
Adele A. Rescigno², and Ugo Vaccaro²

¹ MASCOTTE, joint project CNRS-INRIA-UNSA, F-06902 Sophia-Antipolis, France

² Dipartimento di Informatica, Università di Salerno, 84084 Fisciano (SA), Italy

Abstract. We study algorithmic and complexity issues originating from the problem of data gathering in wireless networks. We give an algorithm to construct minimum makespan transmission schedules for data gathering when the communication graph G is a tree network, the interference range is *any* integer $m \geq 2$, and no buffering is allowed at intermediate nodes. In the interesting case in which all nodes in the network have to deliver an arbitrary non-zero number of packets, we provide a closed formula for the makespan of the optimal gathering schedule. Additionally, we consider the problem of determining the computational complexity of data gathering in general graphs and show that the problem is NP-complete. On the positive side, we design a simple $(1 + 2/m)$ factor approximation algorithm for general networks.

1 Introduction

Technological advances in very large scale integration, wireless networking, and in the manufacturing of low cost, low power digital signal processors, combined with the practical need for real time data collection have resulted in an impressive growth of research activities in Wireless Sensor Networks (WSN). Usually, a WSN consists of a large number of small-sized and low-powered sensors deployed over a geographical area, and of a base station where data sensed by the sensors are collected and accessed by the end user. Typically, all nodes in a WSN are equipped with sensing and data processing capabilities; the nodes communicate with each other by means of a wireless multi-hop networks.

A basic task in a WSN is the systematic gathering at the base station of the sensed data, generally for successive further processing. Due to the current technological limits of WSN, this task must be performed under quite strict constraints. Sensor nodes have low-power radio transceivers and operate with non-replenishable batteries. Data transmitted by a sensor reach only the nodes within the transmission range of the sender. Nodes far from the base station must use intermediate nodes to relay data transmissions. Data collisions, that happen when two or more sensors send data to a common neighbor at the same time, may disrupt the data aggregation process at the base station. An other important factor to take into account when performing data gathering is the

* Funded by ANR AGAPE, ANR GRATTEL and APRF PACA FEDER RAISOM

latency of the information accumulation process. Indeed, the data collected by a node of the network can frequently change, thus it is essential that they are received by the base station as soon as it is possible without being delayed by collisions [18]. The same problem was asked by France Telecom (see [6]) on how to bring internet to places where there is no high speed wired access. Typically, several houses in a village want to access a gateway connected to internet (for example via a satellite antenna). To send or receive data from this gateway, they necessarily need a multiple hop relay routing.

All these issues raise unique challenging problems towards the design of efficient algorithms for data gathering in wireless networks. It is the purpose of this paper to address some of them and propose effective methods for their solutions.

1.1 The Model

We adopt the network model considered in [1, 2, 9, 10, 14]. The network is represented by a node-weighted graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. More specifically, each node in V represents a device that can transmit and receive data. There is a special node $s \in V$ called the *Base Station (BS)*, which is the final destination of all data possessed by the various nodes of the network. Each $v \in V - \{s\}$ has an integer weight $w(v) \geq 0$, that represents the number of data packets it has to transmit to s . Each node is equipped with an half-duplex transmission interface, that is, the node cannot transmit and receive at the same time. There is an edge between two nodes u and v if they can communicate. So $G = (V, E)$ represents the graph of possible communications. Some authors consider that two nodes can communicate only if their distance in the Euclidean space is less than some value. Here we consider general graphs in order to take into account physical or social constraints, like walls, hills, impediments, etc.. In that context paths and trees represent the cases where the communications are done with antennas only in few directions or urban situations with possible communications only along streets. Furthermore, many transmission protocols use a tree of shortest paths for routing.

Time is slotted so that one-hop transmission of a packet (one data item) consumes one time slot; the network is assumed to be synchronous. These hypotheses are strong ones and suppose a centralized view. The values of the completion time we obtain will give lower bounds for the corresponding real life values. Said otherwise, if we fix a value on the completion time, our results will give an upper bound on the number of possible users in the network.

Following [10, 12, 18], we assume that no buffering is done at intermediate nodes and each node forwards a packet as soon as it receives it. One of the rationales behind this assumption is that it might be too much energy consuming to hold data in the node memory; moreover, it also free intermediate nodes from the need to maintain costly state information.

Finally we use a binary model of interference based on the distance in the communication graph. Let $d(u, v)$ denote the distance (that is, the length of a shortest path) between u and v in G . We suppose that when a node u transmits, all nodes v such that $d(u, v) \leq m$ are subject to the interference of u 's

transmission and cannot receive any packet from their neighbors. This model is a simplified version of the reality, where a node is under the interference of all the other nodes and where models based on SNR (Signal-to-Noise Ratio) are used. However our model is more accurate compared to the classical binary model ($m = 1$), where a node cannot receive a packet only in the case one of its neighbor transmits. We suppose all nodes have the same interference range m ; in fact m is only an upper bound on the possible range of interferences since due to obstacles the range can be sometimes lower (however, see also [17] for a critique of this model).

Under above model, simultaneous transmissions among pair of nodes are successful whenever transmission and interference constraints are respected. Namely, a transmission from node v to w is called collision-free if, for all simultaneous transmissions from any node x , it holds: $d(v, w) = 1$ and $d(x, w) \geq m + 1$.

The gathering process is called *collision-free* if each scheduled transmission is collision-free. The collision-free data gathering problem can be stated as follows.

Data Gathering. *Given a graph $G = (V, E)$, a weight function $w : V \rightarrow N$, and a base station s , for each node $v \in V - \{s\}$ schedule the multi-hop transmission of the $w(v)$ data items sensed at node v to base station s so that the whole process is collision-free and the makespan, i.e., the time when the last data item is received by s , is minimized.*

Actually, we will describe the gathering schedule by illustrating the schedule for the equivalent personalized broadcast problem, since this last formulation allows us to use a simpler notation.

Personalized broadcast: *Given a graph G , a weight function $w : V \rightarrow N$, and a BS s , for each node $v \neq s$ schedule the multi-hop transmission from s to v of the $w(v)$ data items destined to v so that the whole process is collision-free and the makespan, i.e., the time when the last data item is received at the corresponding destination node, is minimized.*

We notice that any collision-free schedule for the personalized broadcasting problem is equivalent to a collision-free schedule for data gathering. Indeed, let \mathcal{T} be the last time slot used by a collision-free personalized broadcasting schedule; any transmission from a node v to its neighbor w occurring at time slot k in the broadcasting schedule corresponds to a transmission from w to v scheduled at time slot $\mathcal{T} + 1 - k$ in the gathering schedule. Moreover, if two transmissions in the broadcasting schedule, say from node v to w and from v' to w' , do not collide then $d(v', w)$, $d(v, w') \geq m + 1$; this implies that, in the gathering schedule, the corresponding transmissions from w to v and from w' to v' do not collide either. Hence, if one has an (optimal) broadcasting schedule from s , then one can get an (optimal) solution for gathering at s .

Let S be a personalized broadcasting schedule for the graph G and BS s . We denote by \mathcal{T}_S the makespan of S , i.e., the last time slot in which a packet is sent along *any* edge of the graph. Moreover, we denote by $\mathcal{T}_S(x)$ the time slot in which BS s transmits the last of the $w(x)$ packets destined to node x during

the execution of the schedule S . Clearly, the makespan of S is

$$\mathcal{T}_S = \max \{d_S(s, x) + \mathcal{T}_S(x) \mid x \in V, w(x) > 0\}, \quad (1)$$

where $d_S(s, x)$ is the number of hops used in S for a packet to reach x .

The makespan of an optimal schedule³ is $\mathcal{T}^*(G, s) = \min_S \mathcal{T}_S$, where the minimum is taken over all collision-free personalized broadcasting schedules for the graph G and BS s . When s is clear from the context, we simply write $\mathcal{T}^*(G)$ to denote the optimal makespan value.

1.2 Our Results and Related Work

Our first result is presented in Section 2, where we give an algorithm to determine an optimal transmission schedule for data gathering (personalized broadcasting) in case the communication graph G is a tree network and the interference range is *any* integer $m \geq 2$. Our algorithm works for general weight functions w on the set of nodes V of G . In the interesting case in which the weight function w assume non-zero values on V we are also able to determine a closed formula for the makespan of the optimal gathering schedule. The papers most closely related to our results are [2, 10, 12]. Paper [10] firstly introduced the data gathering problem in a model for sensor networks very similar to the one adopted in this paper. The main difference with our work is that [10] mainly deals with the case where nodes are equipped with directional antennas, that is, only the designated neighbor of a transmitting node receives the signal while its other neighbors can simultaneously and safely receive from different nodes. Under this assumption, [10] gives optimal gathering schedules for trees. Again under the same hypothesis, an optimal algorithm for general networks has been presented in [12] in the case each node has one packet of sensed data to deliver. Paper [2] gives optimal gathering algorithms for tree networks in the same model considered in the present paper, but the authors consider only the particular case of interference range $m = 1$. It is worthwhile to notice that, although our results hold for *general* interference range $m \geq 2$, our algorithms (and analysis thereof) are much cleaner and simpler than those for $m = 1$. In view of our results, it really appears that the case of interference range $m = 1$ has a peculiar behavior, justifying the quite detailed case analysis of [2].

Other related results appear in [1, 4, 5, 7], where fast gathering with omnidirectional antennas is considered under the assumption of possibly different transmission and interference ranges. That is, when a node transmits all the nodes within a fixed distance d_T in the graph can receive, while nodes within distance d_I ($d_I \geq d_T$) cannot listen to other transmissions due to interference (in our paper $d_I = m$ and $d_T = 1$). However, unlike the present paper, all of the above works explicitly allow data buffering at intermediate nodes.

In Section 4, we consider the problem of assessing the hardness of data gathering

³ Note that, by the equivalence between data gathering and personalized broadcasting, in the following we will use $\mathcal{T}^*(G)$ to denote interchangeably the makespan of the data gathering and the personalized broadcasting.

in general graphs and show that the problem is NP-complete. In Section 3 we give a simple $(1 + 2/m)$ factor approximation algorithm for general networks.

Due to space limits, most of the proofs are omitted. The full version of the paper is available on ArXiv.

2 Scheduling in Trees

In this section we describe scheduling algorithms when the network topology is a tree $T = (V, E)$. We first give a polynomial time algorithm for obtaining optimal personalized broadcast schedules in case of strictly positive node weights. Subsequently, in the general case when some nodes can have zero weight, we derive an $O(\delta W^{3\delta})$ algorithm for obtaining an optimal schedule, where W is the sum of the weights of the nodes in the network (number of data packets transmitted) and δ is the BS degree.

Let $T_1, T_2, \dots, T_\delta$ be the subtrees of T rooted at the children of the BS s .

In order to describe the scheduling, we use the following nomenclature.

- *At time t* : During the t -th time slot (one time slot corresponding to a one hop transmission of one packet).
- *Transmit to node v at time t* : a packet to v is sent along a path ($s = x_0, x_1, \dots, x_\ell = v$) from s to v in T starting at time t , that is, the packet is transmitted with a call from x_j to x_{j+1} at step $t + j$, for $j = 0, \dots, \ell - 1$.
- *Node v is completed* (at time t): s has already transmitted all the $w(v)$ packets to v (within some time $t' < t$).
- *Transmit to T_i at time t* : a packet is transmitted at time t to a node v in T_i , where v is chosen as the node having maximum level among all nodes in T_i which are not completed at time t .
- *T_i is completed*: each node v in T_i is completed.

Fact 1 *Let s transmit to a node $u \in V(T_i)$ at time t and to node $v \in V(T_j)$ at time $t' > t$. The calls done during the transmission from s to u and the calls of the transmission from s to v do not interfere if and only if $t' \geq t + \Delta(u, v)$, where the inter-call interval $\Delta(u, v)$ is defined as*

$$\Delta(u, v) = \begin{cases} \min\{d(s, u), m\} & \text{if } i \neq j, \\ \min\{d(s, u), m + 2\} & \text{if } i = j. \end{cases} \quad (2)$$

2.1 Trees with non-zero node weights

In this section we show how to obtain an optimal transmission schedule of the packets to the nodes in a tree T when $w(v) \geq 1$, for each node v in T .

For each subtree T_i of T , for $i = 1, \dots, \delta$, we denote by

- s_i the root of T_i ;
- $|A_i| = \sum_{v \in A_i} w(v)$: the total weight of the nodes in $A_i = \{v \in V(T_i) \mid d(s, v) \leq m\}$, that is, of the nodes in T_i that are at level at most m in T ;
- $|B_i| = \sum_{v \in B_i} w(v)$: the total weight of all the nodes in $B_i = \{v \in V(T_i) \mid d(s, v) = m + 1\}$, that is, of the nodes in T_i that are at level $m + 1$ in T ;

- $|C_i| = \sum_{v \in C_i} w(v)$: the total weight of all the nodes in $C_i = \{v \in V(T_i) \mid d(s, v) \geq m + 2\}$, that is, of the nodes in T_i that are at level $m + 2$ or more in T ;
- $|T_i|$: the total weight of nodes in T_i , that is, $|T_i| = |A_i| + |B_i| + |C_i|$.

Definition 1 Given $i, j = 1, \dots, \delta$ with $i \neq j$, we say that

$$T_i \succeq T_j \text{ if } \begin{cases} |B_i| + |C_i| \geq |B_j| + |C_j| & \text{whenever } |B_i| + |C_i| > 0, \\ |A_i| - w(s_i) \geq |A_j| - w(s_j) & \text{whenever } |B_i| + |C_i| = |B_j| + |C_j| = 0, \\ & |A_i| > w(s_i) \\ w(s_i) \geq w(s_j) & \text{whenever } |T_i| = w(s_i) \text{ and } |T_j| = w(s_j) \end{cases}$$

Theorem 1 Let the interference range be $m \geq 2$. Let T be a tree with node weight $w(v) \geq 1$, for each $v \in V$. Consider T as rooted at the BS s and (w.l.o.g.) let its subtrees be indexed so that $T_1 \succeq T_2 \succeq \dots \succeq T_\delta$. There exists a polynomial time scheduling algorithm S for T such that

$$\mathcal{T}_S = \mathcal{T}^*(T) = \sum_{\substack{u \in V \\ d(s, u) \leq m}} w(u)d(s, u) + m \sum_{i=1}^{\delta} (|B_i| + |C_i|) + M, \quad (3)$$

where

$$M = \max\{0, (|B_1| + |C_1|) - \sum_{i=2}^{\delta} |T_i|, (|B_1| + 2|C_1|) + \sum_{i=2}^{\delta} w(s_i) - 2 \sum_{i=2}^{\delta} |T_i|\} \quad (4)$$

Proof (Sketch). The proof consists in showing that the value in the statement of Theorem 1 is a lower bound on the makespan of any schedule. Subsequently, we prove that the scheduling algorithm given in Figure 1 is collision-free and its makespan matches the lower bound. \square

We notice that in the special case $\delta = 1$, Theorem 1 reduces to

Corollary 1 [10] Let \mathcal{L} be a line with nodes $\{0, 1, \dots, n\}$, BS at node 0, and let $w(\ell) \geq 1$ be the weight of node ℓ , for $\ell = 1, \dots, n$. Then $\mathcal{T}^*(\mathcal{L}) = \sum_{\ell=1}^{m+1} \ell \cdot w(\ell) + (m+2) \sum_{\ell \geq m+2} w(\ell)$.

Example 1. We stress that each of the values of M in (4) is attained by some tree. Figure 2 shows an example for each case assuming the interference range be $m = 3$. The vertices of the trees are labeled with their weights and the subtrees are ordered from left to right according to Definition 1.

a) Consider the tree T in Fig.2 a). T has subtrees T_1, T_2, T_3 with $|B_1| = 3$, $|C_1| = 1$, $|T_2| + |T_3| = 12$ and $w(s_2) + w(s_3) = 2$. Therefore, $|B_1| + |C_1| - (|T_2| + |T_3|) = -8 < 0$ and $|B_1| + 2|C_1| + (w(s_2) + w(s_3)) - 2(|T_2| + |T_3|) = -17 < 0$. Hence, $M = 0$ in this case.

b) Consider the tree T in Fig.2 b). T has subtrees T_1, T_2, T_3 with $|B_1| = 7$, $|C_1| = 3$, $|T_2| + |T_3| = 9$ and $w(s_2) + w(s_3) = 2$. Therefore, $|B_1| + |C_1| - (|T_2| + |T_3|) = 1 > 0$ and $|B_1| + 2|C_1| + (w(s_2) + w(s_3)) - 2(|T_2| + |T_3|) = -3 < 0$. Hence, $M = |B_1| + |C_1| - \sum_{i=2}^{\delta} |T_i|$ in this case.

c) Consider the tree T in Fig.2 c). T has subtrees T_1, T_2, T_3, T_4 with $|B_1| = 2$, $|C_1| = 12$, $|T_2| + |T_3| + |T_4| = 13$ and $w(s_2) + w(s_3) + w(s_4) = 5$. Therefore, $|B_1| + |C_1| - (|T_2| + |T_3| + |T_4|) = 1 > 0$ and $|B_1| + 2|C_1| + (w(s_2) + w(s_3) + w(s_4)) - 2(|T_2| + |T_3| + |T_4|) = 5 > 1$. Hence, $M = |B_1| + 2|C_1| + \sum_{i=2}^{\delta} w(s_i) - 2 \sum_{i=2}^{\delta} |T_i|$.

TREE-scheduling (T, s) [T has non empty subtrees T_1, \dots, T_δ and root s]

Phase 1: Set $\tau = 1$; $previous = 0$; $i = 0$

Set $a_k = |A_k|$, $b_k = |B_k|$, $c_k = |C_k|$, and $n_k = |T_k|$, for $k = 1, \dots, \delta$

Set $D = \{1, \dots, \delta\}$ [D represents the set of indices of subtrees with $n_k > 0$]

Phase 2: while $|D| \geq 2$

$i = i + 1$

Execute the following **Iteration Step i**

Set $\alpha[i] = False$

Let $k \in D - \{previous\}$ be s.t. $T_k \succeq T_j$, for each $j \in D - \{previous\}$ [cfr. Def. 1]

Transmit to T_k at time τ

$n_k = n_k - 1$

(2.1) if $b_k + c_k > 0$ then

if $c_k > 0$ then $\alpha[i] = True$ and $c_k = c_k - 1$ else $b_k = b_k - 1$

$previous = k$

$\tau = \tau + m$

(2.2) if $b_k + c_k = 0$ then

$a_k = a_k - 1$

if $a_k = 0$ then $D = D - \{k\}$

Let u in T_k be the destination of the last transmission by s

$\tau = \tau + d(s, u)$

(2.3) [If the previous transmission was to a node at distance at least $m + 2$ and the actual transmission is to a son of s , we choose an uncompleted son of s different from $s_{previous}$, if any]

if $\alpha[i - 1] = True$ and $d(s, u) = 1$ then

if $|D| \geq 2$ then Transmit to T_h at time τ ,

for some $h \in D - \{previous\}$

$a_h = a_h - 1$

if $a_h = 0$ then $D = D - \{h\}$

$\tau = \tau + 1$

$previous = 0$

Phase 3: Let $D = \{k\}$ [here $|D| = 1$]

while $n_k > 0$

Transmit to T_k at time τ ,

$n_k = n_k - 1$

$\tau = \tau + \min\{d(s, u), m + 2\}$, where u is the destination node in T_k .

Figure 1. The scheduling algorithm on trees.

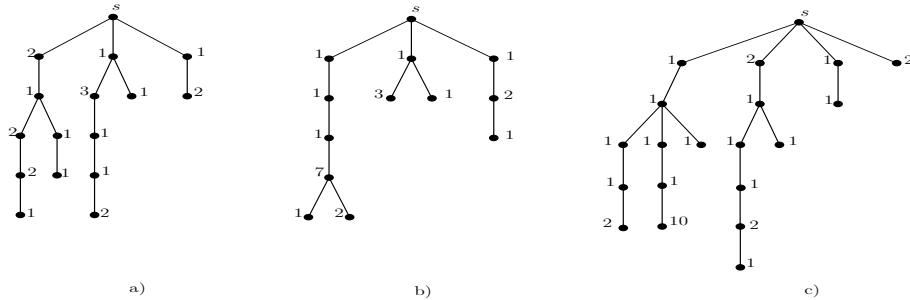


Figure 2. The trees of Example 1.

2.2 Trees with general weight distribution

In this section we present an algorithm for the general case in which only some of the nodes needs to receive packets from the BS s .

Let $T = (V, E)$ be the tree representing the network, and let s be the root of T . Denote by δ the degree of s , and by $T_1, T_2, \dots, T_\delta$ the subtrees of T rooted at the children of s . We present an algorithm which gives an optimal schedule in time $O(\delta W^{3\delta})$, where W is the number of items to be transmitted (i.e, the sum of the weights). However, for sake of simplicity, in the following we limit our analysis to the case $w(v) \in \{0, 1\}$, for each $v \in V - \{s\}$.

Lemma 1 *For each $u, v \in V$, if either of the following conditions hold*

a) $2 \leq d(s, u) < d(s, v) \leq m$

b) $d(s, u) > d(s, v) \geq m + 2$ and $u, v \in V(T_i)$, for some $1 \leq i \leq \delta$,

then there exists an optimal schedule where s transmits to u before than to v .

Based on Lemma 1, we consider the lists C_i, B_i, A_i , for $i = 1, \dots, \delta$, where:

$C_i = (x_{i,1}, x_{i,2}, \dots)$ consists of all the nodes in T_i with $w(x_{i,j}) > 0$ and $d(s, x_{i,j}) \geq m + 2$; nodes are ordered so that $d(s, x_{i,j}) \leq d(s, x_{i,j+1})$ for each $j \geq 1$;

$B_i = (z_{i,1}, z_{i,2}, \dots)$ consists of all the nodes in T_i with $w(z_{i,j}) > 0$ and $d(s, z_{i,j}) = m + 1$; in any order;

$A_i = (y_{i,1}, y_{i,2}, \dots)$ consists of all the nodes in T_i with $w(y_{i,j}) > 0$ and $2 \leq d(s, y_{i,j}) \leq m$; nodes are ordered so that $d(s, y_{i,j}) \geq d(s, y_{i,j+1})$ for $j \geq 1$.

Given integers $c_i \leq |C_i|$, $b_i \leq |B_i|$, $a_i \leq |A_i|$, $r_i \in \{0, 1\}$, for $i = 1, \dots, \delta$, let $S(c_1, \dots, c_\delta, b_1, \dots, b_\delta, a_1, \dots, a_\delta, r_1, \dots, r_\delta)$, denote an optimal schedule satisfying Lemma 1 when the only packets to be transmitted are destined to the first c_i nodes of C_i , b_i nodes of B_i , a_i nodes of A_i , respectively, and, if $r_i = 1$, to the root s_i of T_i , for $i = 1, \dots, \delta$. In the following we will use the compact vectorial notation

$$\mathbf{c} = (c_1, \dots, c_\delta), \quad \mathbf{b} = (b_1, \dots, b_\delta) \quad \mathbf{a} = (a_1, \dots, a_\delta) \quad \mathbf{r} = (r_1, \dots, r_\delta).$$

Therefore, we write $S(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r})$ for $S(c_1, \dots, c_\delta, b_1, \dots, b_\delta, a_1, \dots, a_\delta, r_1, \dots, r_\delta)$. Moreover, let $S(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, t))$ be an optimal schedule satisfying above condition and the additional restriction that the first transmission in the schedule is to a node in T_j where $t \in \{r, C, B, A\}$ specifies whether this node is either the root of T_j , or a node in C_j (by Lemma 1, node x_{j,c_j}), or a node in B_j , or in A_j (by Lemma 1, node y_{j,a_j}).

The makespan of the schedule $S(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r})$ (resp. $S(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, t))$) will be denoted by $\mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r})$ (resp. $\mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, t))$). Clearly,

$$\mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}) = \min_{1 \leq j \leq \delta} \min_{t \in \{r, C, B, A\}} \mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, t)). \quad (5)$$

Denote by \mathbf{e}_i the identity vector $\mathbf{e}_i = (e_{i,1}, \dots, e_{i,\delta})$ with $e_{i,j} = \begin{cases} 1 & \text{if } j = i, \\ 0 & \text{otherwise.} \end{cases}$

The following result is an immediate consequence of Fact 1.

Fact 2 For any $j = 1, \dots, \delta$, it holds

- if $t = r$, then $\mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, r)) = 1 + \mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r} - \mathbf{e}_j)$
- if $t = A$, then $\mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, A)) = d(s, y_{j,a_j}) + \mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a} - \mathbf{e}_j, \mathbf{r})$.
- if $t = B$, i.e., the first transmission is for $z_{j,b_j} \in B_j$, then
 $\mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, B)) =$

$$\min_{k, t'} \begin{cases} m + \mathcal{T}(\mathbf{c}, \mathbf{b} - \mathbf{e}_j, \mathbf{a}, \mathbf{r}, (k, t')) & \text{if } j \neq k \text{ and} \\ & d(s, z_{j,b_j}) \leq \mathcal{T}(\mathbf{c}, \mathbf{b} - \mathbf{e}_j, \mathbf{a}, \mathbf{r}, (k, t')) + m \\ m + 1 + \mathcal{T}(\mathbf{c}, \mathbf{b} - \mathbf{e}_j, \mathbf{a}, \mathbf{r}, (k, t')) & \text{if } j = k \text{ and} \\ & d(s, z_{j,b_j}) \leq \mathcal{T}(\mathbf{c}, \mathbf{b} - \mathbf{e}_j, \mathbf{a}, \mathbf{r}, (k, t')) + m + 1 \\ d(s, z_{j,b_j}) & \text{otherwise} \end{cases}$$

- if $t = C$, i.e., the first transmission is for $x_{j,c_j} \in C_j$, then
 $\mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, C)) =$

$$\min_{k, t'} \begin{cases} m + \mathcal{T}(\mathbf{c} - \mathbf{e}_j, \mathbf{b}, \mathbf{a}, \mathbf{r}, (k, t')) & \text{if } j \neq k \text{ and} \\ & d(s, x_{j,c_j}) \leq \mathcal{T}(\mathbf{c} - \mathbf{e}_j, \mathbf{b}, \mathbf{a}, \mathbf{r}, (k, t')) + m \\ m + 2 + \mathcal{T}(\mathbf{c} - \mathbf{e}_j, \mathbf{b}, \mathbf{a}, \mathbf{r}, (k, t')) & \text{if } j = k \text{ and} \\ & d(s, x_{j,c_j}) \leq \mathcal{T}(\mathbf{c} - \mathbf{e}_j, \mathbf{b}, \mathbf{a}, \mathbf{r}, (k, t')) + m + 2 \\ d(s, x_{j,c_j}) & \text{otherwise} \end{cases}$$

An optimal schedule for T is $S(T) = S(\mathbf{c}_T, \mathbf{b}_T, \mathbf{a}_T, \mathbf{r}_T)$, where $(\mathbf{c}_T, \mathbf{b}_T, \mathbf{a}_T, \mathbf{r}_T)$ includes all the packets in T . In order to obtain the optimal solution we compute the various partial solutions for $(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, t))$; starting from $T(\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (j, t)) = 0$, for each j and t , where $\mathbf{0} = (0, \dots, 0)$ is the null vector.

We know that $c_k + b_k + a_k \leq \sum_{v \in V} w(v) = W$ and $r_k \in \{0, 1\}$, for $k = 1, \dots, \delta$; moreover, the pair (j, t) can assume at most 4δ values. Therefore, since $w(v) \in \{0, 1\}$ for each $v \in V$, we get $W \leq |V|$ and the number of different values we need to compute is $O(\delta|V|^{3\delta})$.

For general weights, each node v needs to appear in the proper list (among A_i, B_i , and C_i , for $i = 1, \dots, \delta$) with multiplicity equal to $w(v)$. Hence, our result assumes the following form.

Theorem 2 It is possible to obtain an optimal schedule in time $O(\delta W^{3\delta})$.

3 General Topologies

We present an algorithm for Personalized Broadcasting in general graphs and prove that it achieves an approximation ratio of $1 + \frac{2}{m}$, where m is the interference range. We then show that if one requires that the personalized broadcasting has to be done using a routing tree, then the problem is NP-complete. We stress that this practical requirement is widely adopted, indeed it avoids that intermediate nodes have to forward data in a way that depends on source and destination information. The same scenario for $m = 1$ is considered in [9].

3.1 The approximation algorithm

Consider an arbitrary topology graph $G = (V, E)$ with BS s and node weight $w(v) \geq 0, v \in V - \{s\}$. Let SP be a set of shortest paths from s to each node in $V - \{s\}$. We route transmissions along the paths in SP .

```

Graph-SPschedule( $G, SP, s$ )
Set  $t = 1; h = \max_{u \in V} d(s, u)$ 
Set  $w_\ell = \sum_{v \in V, d(s, v) = \ell} w(v)$ , for  $\ell = 1, \dots, h$ 
while  $\sum_\ell w_\ell > 0$ 
    Let  $L = \max\{\ell | w_\ell > 0\}$ 
    Establish an (arbitrary) ordering on the  $w_L$  packets to be transmitted
    to nodes at distance  $L$  from  $s$ 
    For  $j = 1$  to  $w_L$ 
         $s$  transmits at time  $t$  the  $j$ -th data packet in the above ordering
         $t = t + \min\{L, m + 2\}$ 
     $w_L = 0$ 

```

Figure 3. The general graphs scheduling algorithm.

Lemma 2 *The makespan of the schedule produced by Graph-SPschedule(G, SP, s) is*

$$\max \left\{ \sum_{\substack{v \in V \\ d(s, v) \leq m+1}} w(v)d(s, v) + (m+2) \sum_{\substack{v \in V \\ d(s, v) \geq m+2}} w(v), \max_{\ell \geq m+2} \left\{ \ell - m - 2 + (m+2) \sum_{\substack{v \in V \\ d(s, v) \geq \ell}} w(v) \right\} \right\}.$$

The analysis of the algorithm would be very simple if we had to deal only with trees (indeed schedules with optimal makespan for trees are given in Sections 2). However, even if we restrict ourselves to packets transmission on a (shortest path) tree, we *still* need to deal with possible collisions due to the edges in $E - E(SP)$. In order to see that our algorithm does not suffer from interferences, let us first notice that if $(u, v) \in E$ then $|d(s, u) - d(s, v)| \leq 1$. Moreover, if s transmits to u at time t and to v at time $t' > t$ then the Graph-SPschedule algorithm imposes that $t' = t + \min\{d(s, u), m + 2\}$. By this, as in Fact 1, we get that no collision occurs during the execution of Graph-SPschedule.

Theorem 3 *Let $G = (V, E)$ be a graph with BS $s \in V$ and $w(u) \geq 0$, for each $u \in V - \{s\}$, and let the interference range be m . The makespan \mathcal{T} of the schedule produced by Graph-SPschedule(G, SP, s) satisfies*

$$\frac{\mathcal{T}}{\mathcal{T}^*(G)} \leq 1 + \frac{2}{m},$$

where $\mathcal{T}^*(G)$ is the makespan of an optimal scheduling for G .

4 Complexity Results

We now show that the Data Gathering Problem is NP-complete if the process must be performed along the edges of a *routing tree*.

Our proof assumes $m \geq 2$. The case $m = 1$ is claimed to be NP-complete in [9]; however the proof is incorrect. Firstly, it uses invalid results concerning trees. Indeed the authors claim that in the case $m = 1$, a tree with n vertices and weight 1 in each node has makespan equal to $3n - 2$. As a counterexample, consider the tree formed by δ paths of length 2 sharing the node s , so with $n = 2\delta + 1$ nodes. The makespan in this case is $2\delta = n - 1$ (see [2] for exact values for trees). As a matter of fact, the value in [9] is true only for paths with BS at one end. Additionally, one can easily see that the reduction employed in [9] is, in general, not computable in polynomial time.

To prove our NP-completeness result, let us consider the decision version of the equivalent Minimum Time Personalized Broadcasting.

MTPB (Minimum Time Personalized Broadcasting)

Instance: A graph $G = (V, E)$, an interference range m , a special node $s \in V$, integer weights $w(v) \geq 0$ for $v \in V - \{s\}$, and an integer bound K .

Question: Is there a routing tree in G and a multi-hop schedule on it of the $w(v)$ packets from s to node v , for each $v \in V$, so that the process is collision-free and the makespan is $\mathcal{T} \leq K$?

We show now that **MTPB** is NP-complete. It is clearly in NP. We prove its NP-hardness by a reduction from the well known Partition Problem [13].

PARTITION

Instance: $n + 1$ integers a_1, a_2, \dots, a_n, B such that $\sum_{i=1}^n a_i = 2B$.

Question: Is there a subset $S \subset \{1, 2, \dots, n\}$ such that $\sum_{i \in S} a_i = B$?

Given a **PARTITION** instance, we construct a **MTPB** instance as follows:

- The graph is $G = (V, E)$ with node set

$$V = \{s\} \cup \{u_j^0, v_j^0 \mid 1 \leq j \leq m + n + 1\} \cup \{u_j^i, v_j^i \mid 1 \leq i \leq n, 0 \leq j \leq m\} \\ \cup \{x^i \mid 1 \leq i \leq n\},$$

edge set

$$E = \{(s, u_1^0), (s, v_1^0)\} \cup \{(u_j^0, u_{j+1}^0), (v_j^0, v_{j+1}^0) \mid 1 \leq j \leq m + n\} \\ \cup \{(u_{m+n+1}^0, u_0^1), (v_{m+n+1}^0, v_0^1)\} \\ \cup \{(u_j^i, u_{j+1}^i), (v_j^i, v_{j+1}^i) \mid 1 \leq i \leq n, 0 \leq j \leq m - 1\} \\ \cup \{(u_1^i, u_0^{i+1}), (v_1^i, v_0^{i+1}) \mid 1 \leq i \leq n - 1\} \\ \cup \{(u_m^i, x^i), (v_m^i, x^i) \mid 1 \leq i \leq n\};$$

and node weights

$$w(u_j^0) = w(v_j^0) = 0, \text{ for } j = 1 \dots, m + 1, \\ w(u_j^0) = w(v_j^0) = 1, \text{ for } j = m + 2 \dots, m + n + 1, \\ w(u_j^i) = w(v_j^i) = 0, \text{ for } i = 1 \dots, n \text{ and } j = 0 \dots, m, \\ w(x^i) = a_i, \quad \text{for } i = 1 \dots, n.$$

- The interference parameter is a fixed integer $m \geq 2$;
- The bound is $K = 2m(B + n) + 2$.

We notice that the graph G can be constructed in polynomial-time. Moreover, it can be shown that the **PARTITION** instance admits an answer “Yes” if and only if there exists a schedule for the **MTPB** instance such that the makespan is $\mathcal{T} \leq K$. Hence we get

Theorem 4 *The MTPB problem is NP-complete.*

References

1. Bermond J-C., Galtier J., Klasing R., Morales N., Pérennes S.: Hardness and approximation of gathering in static radio networks. PPL 16 (2), 165–183 (2006).
2. Bermond J-C., Gargano L., Rescigno A.: Gathering with Minimum Delay in Sensor Networks. Proc. SIROCCO 2008, LNCS 5058, 262–276 (2008).
3. Bermond J-C., Correa R., Yu M.-L.: Optimal Gathering Protocols on Paths under Interference Constraints. Discrete Mathematics 309(18), 5574–5587 (2009).
4. Bermond J-C., Peters J.: Efficient gathering in radio grids with interference. Proc. AlgoTel’05, Presqu’île de Giens, 103–106 (2005).
5. Bermond J-C., Yu M.-L.: Optimal gathering algorithms in multi-hop radio tree networks with interferences. Ad Hoc and S.W.N., 9 (1–2), 109–128 (2010).
6. Bertin P., Bresse J-F., Le Sage B.: Accès haut débit en zone rurale: une solution “ad hoc”. France Telecom R&D 22, 16–18 (2005).
7. Bonifaci V., Korteweg P., Marchetti-Spaccamela A., Stougie L.: An Approximation Algorithm for the Wireless Gathering Problem. Op. Res. Lett. 36 (5), 605–608 (2008).
8. Bonifaci V., Klasing R., Korteweg P., Marchetti-Spaccamela A., Stougie L.: Data Gathering in Wireless Networks. Graphs and Algorithms in Communication Networks, A.Koster and X. Munoz editors, Springer Monograph, 357–377 (2010).
9. Choi H., Wang J., Hughes E.A.: Scheduling for information gathering on sensor network. Wireless Network 15, 127–140 (2009).
10. Florens C., Franceschetti M., McEliece R.J.: Lower Bounds on Data Collection Time in Sensory Networks. IEEE J. on SAC. 22 (6), 1110–1120 (2004).
11. Gargano L.: Time Optimal Gathering in Sensor Networks. Proc. SIROCCO 2007, LNCS 4474, 7–10 (2007).
12. Gargano L., Rescigno A.A.: Optimally Fast Data Gathering in Sensor Networks. Discrete Applied Mathematics 157, 1858–1872 (2009).
13. Garey M. R., Johnson D. S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
14. Gasieniec L., Potapov I.: Gossiping with Unit Messages in Known Radio Networks. IFIP TCS, 193–205 (2002).
15. Pelc A.: Broadcasting in radio networks. Handbook of Wireless Networks and Mobile Computing, I. Stojmenovic, Ed. John Wiley and Sons, Inc., 509–528 (2002).
16. Revah Y., Segal M.: Improved bounds for data-gathering time in sensor networks. Computer Communications 31(17), 4026–4034 (2008).
17. Schmid S., Wattenhofer R.: Algorithmic models for sensor networks. IPDPS 2006
18. X. Zhu, Tang B., Gupta H.: Delay efficient data gathering in sensor networks, Proc. of MSN 2005, LNCS 3794, 380–389 (2005).