Optimal Time Data Gathering in Wireless Networks with Multidirectional Antennas

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

Optimal Time Data Gathering in Wireless Networks with Multidirectional Antennas

Jean–Claude Bermond*
MASCOTTE
joint project CNRS-INRIA-UNSA
06902 Sophia-Antipolis, France

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

Stephane Perénnes
MASCOTTE
joint project CNRS-INRIA-UNSA
06902 Sophia-Antipolis, France

Adele A. Rescigno
Dipartimento di Informatica
Università di Salerno
84084 Fisciano (SA), Italy

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

Abstract

A Wireless Network consists of a large number of devices, deployed over a geographical area, and of a base station where data sensed by the devices are collected and accessed by the end users. In this paper 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 under the following hypotheses: the communication graph G is a tree network, the transmissions in the network can interfere with each other up to distance m, where $m \geq 2$, 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.

Keywords: Data gathering, personalized broadcasting, multidirectional antennas, sensor networks, radio networks, interference.

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

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 users. 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 network.

A basic task in a WSN is the systematic gathering at the base station of the sensed data, generally for 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 gathering process. Another important factor to take into account when performing data gathering is the *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 [16]. The same problem was posed 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 the 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 a 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. In fact one gets a symmetric digraph (the transmissions are directed), which is modeled by an undirected graph. 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. Simple graphs modeling urban situations are paths, stars, and grids. Although they are not representative of real networks, we study in this article trees as they contain paths and stars as a special cases and they are the first cases where the complexity of the gathering problem is unknown. Furthermore, many protocols of transmission use a tree of shortest paths for routing.

Time is slotted so that a 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, 16], 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 frees 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 neighbors 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 [15] for a critique of this model).

Under the above model, simultaneous transmissions among pairs 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, the following holds:

$$d(v, w) = 1, \qquad d(x, w) \geq m + 1.$$

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

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

1.2 Gathering vs. Personalized Broadcasting

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

> **Personalized broadcast:** *Given a graph G, a weight function $w : V \to N$, and a BS s, for each node $v \neq s$ schedule the multi-hop transmission from s to v of the $w(v)$ packets destined to v so that the whole process is collision–free and the makespan, i.e., the time when the last packet 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. As the graph is symmetric, when two transmissions in the broadcasting schedule from x to y and v to w do not interfere (can be done simultaneously), that means that $d(x, w) \geq m + 1$ and $d(v, y) \geq m + 1$ and so the reverse transmissions in the gathering schedule do not interfere. Hence, if one has an (optimal) broadcasting schedule from s, then one can have an (optimal) solution for gathering at s.

Let S be a personalized broadcasting schedule for the graph G and the Base Station 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 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

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

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

The makespan of an optimal schedule 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 the BS s. When s is clear from the context, we simply write $\mathcal{T}^*(G)$ to denote the optimal makespan value.

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 of the personalized broadcasting.

1.3 Our Results and Related Work

In this paper we study algorithmic and complexity issues related to the problem of data gathering or personalized broadcasting. Our first main result is presented in Section 2, where we give algorithms 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 $m \geq 2$. In the interesting case in which the weight function w assumes non-zero values on V we are also able to determine a closed formula for the makespan of the optimal gathering schedule. Under the assumption that the weight function w can also assume value zero at some of the nodes in V, we present a pseudo–polynomial dynamic programming algorithm which outputs an optimal schedule in time $O(\delta W^{4\delta})$, where δ is the degree of the BS and $W = \sum_{v \in V-\{s\}} w(v)$ is the total number of items to be transmitted. 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 unidirectional 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 only for 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 detailed case analysis of [2].

Other related results appear in [1, 3, 4, 5, 7] (see [8] for a survey), where fast gathering with multidirectional 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 the case of tree networks a solution is given for $m = 1$ with buffering in [5]. The values obtained for the makespan are smaller when buffering is allowed. To see that consider the case $m = 1$ and the tree consisting of the source s, with one child s_1 and two branches s_1, u_1, v_1 and s_1, u_2, v_2 and suppose the source has to transmit one message to v_1 and one to v_2. If no buffering is allowed the source s will transmit the message to v_1 at time 1 and it will arrive at time 3 and due to interference s can transmit to v_2 only at time 4, the message arriving at time 6. If buffering is allowed the source can transmit at time 1 to s_1 the message for v_1; s_1 will transmit it at time 2 to u_1; at time 3 the source sends the message for v_2 to s_1; at time 4 we have two non-interfering calls (u_1, v_1) and (s_1, u_2) and the message for v_1 is arrived and finally at time 5 u_2 transmits to v_2 and the makespan is $5 < 6$.

In Section 4, we consider the problem of assessing the hardness of data gathering in general graphs and show that the problem is NP–complete. We also give in Section 3 a simple $(1+2/m)$ factor approximation algorithm for general networks.

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 the case of strictly positive node weights. Subsequently, in the general case when some nodes can have zero weight, we derive an $O(\delta W^{4\delta})$ algorithm for obtaining an optimal schedule, where W is the sum of the weights of the nodes in the network (number of data packets to be transmitted) and δ is the BS degree.

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

Definition 1. *We use the following notation.*

1. At time t: *During the t-th time slot (one time slot corresponding to a one hop transmission of one packet).*
2. Transmit to node v at time t: *a packet to v is sent along a path $(s = x_0, x_1, \cdots, x_\ell = v)$ 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 $j = 0, \cdots, \ell - 1$.*
3. *Node v* is completed *(at time t): s has already transmitted all the $w(v)$ packets to v (within some time $t' < t$).*
4. 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 a node having maximum level** *among all nodes in T_i which are not completed at time t.*
5. 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*

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

Proof. Let s transmit to $u \in V(T_i)$ at time t and to $v \in V(T_j)$ at time $t' = t + \ell$, for some $\ell > 0$. Denote by $s = u_0, u_1, \cdots, u_k = u$ the path in T from s to u. At time $t + \ell$, the packet for u is transmitted by u_ℓ at level ℓ to its son $u_{\ell+1}$ in T_i, for each $\ell < k = d(s, u)$.

Assume first that $\ell < \Delta(u, v)$. By definition of $\Delta(u, v)$, we have that $\Delta(u, v) \leq d(s, u)$ and the interference range of u_ℓ includes the son of s in T_j (since this last node is at distance at most m from u_ℓ in both cases $i = j$ or $i \neq j$). Hence, an interference occurs between the call from u_ℓ to $u_{\ell+1}$ and the call from s to the root of T_j. Therefore, we need that $t' = t + \ell \geq t + \Delta(u, v)$ must hold.

On the contrary, assume now that $t' = t + \ell \geq t + \Delta(u, v)$. At time t' the packet for u has either already reached its destination u or it has reached a node at distance at least $m + 2$ from s, if $i = j$, or at distance at least m from s, if $i \neq j$; so in both cases it has reached a node at distance at least $m + 1$ from the son of s in T_j. Hence there is no interference between the two calls done at time t'. Since the distance between the endpoints of the calls done at any time $t'' > t'$ does not decrease, subsequent calls done for the transmissions of the packets destined to u and v do not interfere. □

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 $i = 1, \ldots, \delta$, we denote by

- s_i the root of T_i;
- $|A_i| = \sum_{v \in A_i} w(v)$: the total weight of all the nodes in the set $A_i = \{v \in V(T_i) \mid d(s, v) \leq m\}$, 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 2. *Given $1 \leq i, j \leq \delta$ with $i \neq j$, we say that*

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

Notice that it is possible that both $T_i \succeq T_j$ and $T_j \succeq T_i$. This holds if: $|B_i| + |C_i| = |B_j| + |C_j| > 0$, or $|B_i| + |C_i| = |B_j| + |C_j| = 0$ and $|A_i| - w(s_i) = |A_j| - w(s_j) > 0$, or $|T_i| = w(s_i) = w(s_j) = |T_j|$.
We shall prove that

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 $T_1 \succeq T_2 \succeq \ldots \succeq T_\delta$. There exists a polynomial time scheduling algorithm S for T such that*

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

where

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

We notice that in the special case $\delta = 1$, Theorem 1 reduces to the known result for the line:

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

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

a) Consider the tree T in Fig.1 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.1 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.1 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, $M = |B_1| + 2|C_1| + \sum_{i=2}^{\delta} w(s_i) - 2\sum_{i=2}^{\delta}|T_i|$ in this case.

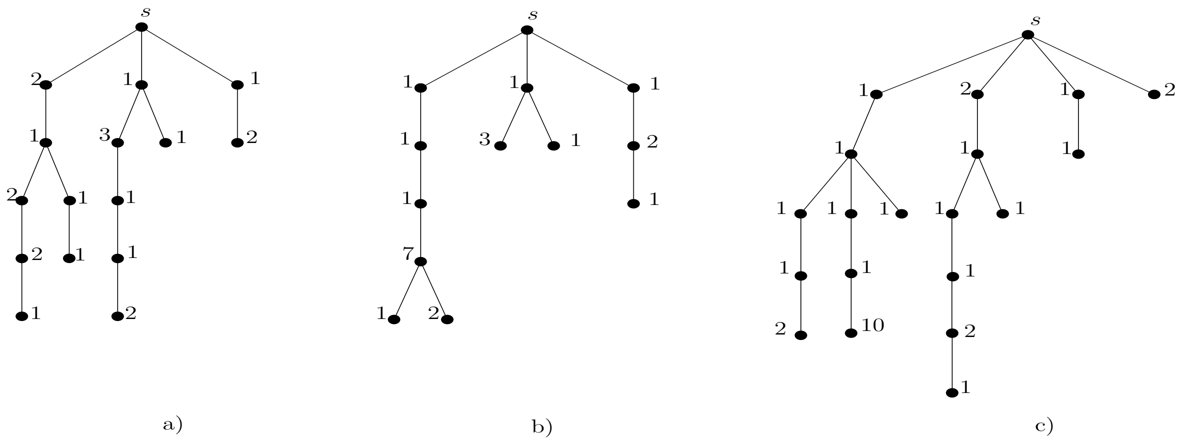


Figure 1.

2.1.1 The lower bound

We first show that the value in Theorem 1 is a lower bound on the makespan of any schedule; next we give an algorithm matching such a bound.

Lemma 1. *Let $T_1 \succeq T_2 \succeq \ldots \succeq T_\delta$ and M be as defined in (4). If $w(v) \geq 1$, for each $v \in V$, then*

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

Proof. By Fact 1, we know that

– when the BS s transmits a packet to any node u with $d(s,u) \leq m$, then at least $d(s,u)$ time slots must elapse before s can transmit a new packet.

– when the BS s transmits a packet to any node u with $d(s,u) > m$, then at least m time slots must elapse before s can transmit a new packet.

Hence, we have that

$$\mathcal{T}^*(T) \geq \sum_{\substack{u \in V \\ d(s,u) \leq m}} w(u)d(s,u) + m\sum_{i=1}^{\delta}(|B_i| + |C_i|). \tag{5}$$

We show now that if $M > 0$ then M additional time slots are necessary.

- Case 1: $M = |B_1| + |C_1| - \sum_{i=2}^{\delta} |T_i| > 0$.
  In this case s must transmit at least $(|B_1| + |C_1|) - \sum_{i=2}^{\delta} |T_i|$ times to nodes at level at least $m + 1$ in T_1 without interleaving any of such transmissions with transmissions to nodes in other subtrees. This implies that, after each of these $(|B_1| + |C_1|) - \sum_{i=2}^{\delta} |T_i|$

transmissions, each inter–call interval (see Fact 1) is either $m+1$ or $m+2$ (that is, at least 1 more than the value accounted in (5)). Hence, the makespan is lower bounded by

$$\sum_{\substack{u\in V\\ d(s,u)\leq m}} w(u)\, d(s,u) + m\sum_{i=1}^{\delta}(|B_i|+|C_i|) + (|B_1|+|C_1|) - \sum_{i=2}^{\delta}|T_i|.$$

- Case 2: $M = (|B_1|+2|C_1|) + \sum_{i=2}^{\delta} w(s_i) - 2\sum_{i=2}^{\delta}|T_i| > 0$.
  We first notice that in this case $|B_1|+|C_1| - \sum_{i=2}^{\delta}|T_i| \leq (|B_1|+2|C_1|) + \sum_{i=2}^{\delta} w(s_i) - 2\sum_{i=2}^{\delta}|T_i|$, which implies $|C_1| \geq \sum_{i=2}^{\delta}|T_i| - \sum_{i=2}^{\delta} w(s_i)$. The BS s has to transmit

  – $|C_1|$ packets to nodes in T_1, each at level at least $m+2$;

  – $|B_1|$ packets to nodes at level $m+1$ in T_1.

  Furthermore, the above transmissions to T_1 can be interleaved only with

  – $\sum_{i=2}^{\delta}|T_i| - \sum_{i=2}^{\delta} w(s_i)$ transmissions to nodes at distance at least 2 and

  – $\sum_{i=2}^{\delta} w(s_i)$ transmissions to nodes at level 1

  in $T_2,\dots T_\delta$. From this and according to Fact 1, we get that in any schedule

  – at least $|C_1| - (\sum_{i=2}^{\delta}|T_i| - \sum_{i=2}^{\delta} w(s_i))$ transmissions to T_1 require an inter–call interval equal to $m+2$ (instead of m as counted in (5)) and

  – at least $|B_1|$ transmissions to T_1 require an inter–call interval equal to $m+1$ (instead of m as counted in (5)).

  Of the above, $\sum_{i=2}^{\delta} w(s_i)$ time slots could be used for the one–hop transmissions of packets to the roots of $T_2,\dots T_\delta$ (interleaving them with the transmissions to T_1). Therefore, with respect to (5), at least

$$2\left[|C_1| - \left(\sum_{i=2}^{\delta}|T_i| - \sum_{i=2}^{\delta} w(s_i)\right)\right] + |B_1| - \sum_{i=2}^{\delta} w(s_i)$$

  additional time slots are necessary. Hence, the makespan is lower bounded by

$$\sum_{\substack{u\in V\\ d(s,u)\leq m}} w(u)\, d(s,u) + m\sum_{i=1}^{\delta}(|B_i|+|C_i|) + |B_1| + 2|C_1| - 2\sum_{i=2}^{\delta}|T_i| + \sum_{i=2}^{\delta} w(s_i).$$

□

2.1.2 The algorithm

Unless otherwise stated, in the following we assume that the subtrees are indexed according to the ranking given in Definition 2, that is $T_1 \succeq \cdots \succeq T_\delta$ and we use the notation of Definition 1. The algorithm is given in Figure 2.

Following is an informal description of the behavior of the algorithm. Consider a generic step i: If there are at least two different uncompleted subtrees, the BS s transmits to the one of them bearing in mind the subtree T_{prev} to which it has transmitted during the previous step $i-1$. In particular, if s has transmitted to a node at distance at least $m+1$ in T_{prev} at step $i-1$, then s transmits to the subtree $T_k \neq T_{prev}$ that comes first in the ordering of the uncompleted subtrees of T at the current step (cfr. Definition 2). Furthermore, if s has transmitted to a node at distance at least $m+2$ in T_{prev} at the previous step and the root of T_k is the only node to be still completed in T_k, then s transmits to T_k (i.e., to the root s_k of T_k) and immediately after to the root s_h of some subtree T_h, where $h \neq prev$ (it can be that $h = k$)[1]. Finally, if only one uncompleted subtree remains then s transmits to one of its farthest nodes (cfr. 4. of Definition 1).

[1]Notice that since, at the current step, T_k is the first subtree in the ordering, with the possible exception of T_{prev}, and since the only node to be completed in T_k is its root then, by Definition 2, the root of each other subtree T_h, where $h \neq prev$, is the only node to be completed in T_h.

TREE-scheduling (T, s) *[T has non-empty subtrees $T_1, \ldots, T_\delta$ and root s]*

Phase 1: Set $\tau = 1$; $prev = 0$;

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

Set $\alpha = False$ *[α is set to $True$ when a transmission to a node at distance at least $m + 2$ occurs]*

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

Set $i = 0$

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

$i = i + 1$

Execute the following **Iteration Step** i

Let $k \in D$ be s.t. $k \neq prev$ and $T_k \succeq T_j$, for each $j \in D$ and $j \neq prev$ *[cfr. Def. 2]*

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** $\alpha = True$ and $c_k = c_k - 1$

else $\alpha = False$ and $b_k = b_k - 1$

$prev = k$

$\tau = \tau + m$

(2.2) **else** *[if $b_k + c_k = 0$]*

$a_k = a_k - 1$

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

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

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

(2.3) *[If the previous transmission was to a node at distance at least $m + 2$ (i.e., $\alpha = True$), and if the current transmission is to the son s_k of s in T_k then we transmit again to an uncompleted son of s, if any different from s_{prev} exists]*

if $\alpha = True$ and $d(s, u) = 1$ **then**

if $|D| \geq 2$ **then** Transmit to T_h at time τ, for some $h \in D$ and $h \neq prev$

$a_h = a_h - 1$

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

$\tau = \tau + 1$

$prev = 0, \ \ \alpha = False$

Phase 3: [*here* $|D| = 1$]

Let $D = \{k\}$

(3.1) **while** $n_k > 0$

Transmit to T_k at time τ,

let u be the destination node

$n_k = n_k - 1$

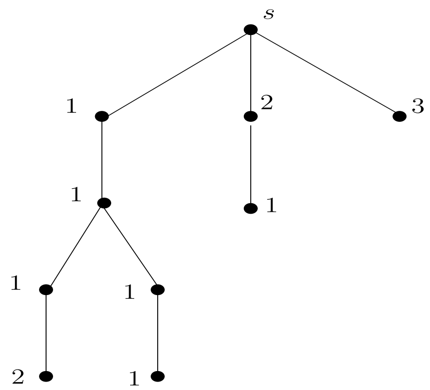
$\tau = \tau + \min\{d(s, u), m + 2\}$.

Figure 2. The scheduling algorithm on trees.

Example. *Consider the tree T in Fig. 3 with BS s. Assume the interference range is $m = 2$. The vertices of T are labeled with their weights and the subtrees are ordered from left to right according to Definition 2. The table in Fig 3 reports the transmissions of s to the subtrees of T*

according to the algorithm TREE-scheduling(T, s).
Each row of the table shows for each iteration step i: The ordering of the subtrees at the beginning of step i (if all the transmissions to some subtree have been made then the subtree does not appear in the ordering); the time τ when s transmits at step i; the index k of the subtree to which s transmits at time τ; the algorithm's point corresponding to the step i; the updated values of prev, α, a_j, b_j, c_j for $j = 1, 2, 3$. At the beginning of the algorithm we have $a_1 = 2$, $b_1 = 2$, $c_1 = 3$, $a_2 = 3$, $a_2 - w(s_2) = 1$, $b_2 = c_2 = 0$, $a_3 = 3$, $a_3 - w(s_3) = 0$, $b_3 = c_3 = 0$.

Note that $\mathcal{T}^*(T) = 21$ as given by Theorem 1. Indeed $\sum_{\substack{u \in V \\ d(s,u) \leq 2}} w(u)d(s,u) = 6 + 4 = 10$; $m \sum_{i=1}^{3}(|B_i| + |C_i|) = 10$ and $M = |B_1| + 2|C_1| + \sum_{i=2}^{3} w(s_i) - 2\sum_{i=2}^{3} |T_i| = 1$.



step i	subtrees' ordering	algorithm's point	τ	k	*prev*	α	a_1	b_1	c_1	a_2	b_2	c_2	a_3	b_3	c_3
					0	*False*	2	2	3	3	0	0	3	0	0
1	$T_1 \succ T_2 \succ T_3$	2.1	1	1	1	*True*	2	2	2	3	0	0	3	0	0
2	$T_1 \succ T_2 \succ T_3$	2.2	3	2	0	*False*	2	2	2	2	0	0	3	0	0
3	$T_1 \succ T_3 \succ T_2$	2.1	5	1	1	*True*	2	2	1	2	0	0	3	0	0
4	$T_1 \succ T_3 \succ T_2$	2.2 2.3	7 8	3	0	*False*	2	2	1	2	0	0	2 1	0	0
5	$T_1 \succ T_2 \succ T_3$	2.1	9	1	1	*True*	2	2	0	2	0	0	1	0	0
6	$T_1 \succ T_2 \succ T_3$	2.2 2.3	11 12	2	0	*False*	2	2	0	1 0	0	0	1	0	0
7	$T_1 \succ T_3$	2.1	13	1	1	*False*	2	1	0	0	0	0	1	0	0
8	$T_1 \succ T_3$	2.2	15	3			2	1	0	0	0	0	0	0	0
9	T_1	3.1	16	1			2	0	0	0	0	0	0	0	0
10	T_1	3.1	19	1			1	0	0	0	0	0	0	0	0
11	T_1	3.1	21	1			0	0	0	0	0	0	0	0	0

Figure 3 : The TREE-scheduling algorithm running, step by step, on the tree in the picture

We first prove that the scheduling produced by the TREE-scheduling algorithm is collision-free. Notice that the arguments in the last case of the proof of Lemma 2 do not hold in the particular case $m = 1$, therefore our results do not extend to the model considered in [2].

Lemma 2. *The scheduling produced by the TREE-scheduling algorithm is collision-free.*

Proof. Suppose that the BS s transmits at time t a packet to a node u, and at time $t' > t$ to a node v. We show that $t' \geq t + \Delta(u,v)$; this, by Fact 1, implies that the lemma holds.
Consider first Phase 3, that is, when $|D| = 1$. In this case, there is exactly one subtree, say T_k, not yet completed, that is, such that $n_k > 0$. If s transmits at time t a packet to a node u in T_k then s transmits again to some v in T_k at time $t + \min\{d(s,u), m+2\} = t + \Delta(u,v)$.
Consider now Phase 2 (here, $|D| \geq 2$).

- If $d(s,u) \leq m$ then, by subphase (2.2) we know that the next transmission by s is at time $t + d(s,u) = t + \Delta(u,v)$, independently from v.

- If $d(s,u) \geq m+1$ then, by subphase (2.1) we know that s transmits again at time $t+m$ to a node v in T_j with $j \neq k$, hence $t + m = t + \Delta(u,v)$.
  Moreover, s transmits again to a node w in T_k, at a time t'' such that

$$t'' \geq t + m + \begin{cases} \Delta(v,w) & \text{if } \; d(s,v) \geq 2 \text{ or } (d(s,v) = 1 \text{ and } d(s,u) = m+1) \\ 2 & \text{otherwise, e.g. if the condition of subphase (2.3) apply.} \end{cases} \tag{6}$$

  In each case $t'' \geq t + \Delta(u,w)$

□

It remains to prove that the algorithm is optimal with respect to the makespan, that is, the last time slot in which the algorithm schedules a call in T matches the lower bound. The assumption that each node has non-zero weight and the order in which BS s schedules the transmissions to the various nodes of each subtree imply that: s calls one of its children during the last time slot in which the algorithm schedules a call in T. For convenience, we formalize that as a lemma.

Lemma 3. *Let t denote the largest integer such that s transmits at time t (to any node) according to the TREE–scheduling algorithm. The makespan of the TREE–scheduling algorithm is t.*

By the above Lemma, we need to show that the largest t such that s transmits at time t is upper bounded by $\sum_{u:d(s,u)\leq m} w(u)\, d(s,u) + m \sum_{i=1}^{d}(|B_i| + |C_i|) + M$. The following Lemmas 4 and 5 prove the optimality of the algorithm and give an upper bound which meets the lower bound of Lemma 1 and so we get Theorem 1. We distinguish two cases according to the values of $M = \max\{0,\ (|B_1|+|C_1|) - \sum_{i=2}^{\delta}|T_i|,\ (|B_1|+2|C_1|) + \sum_{i=2}^{\delta} w(s_i) - 2\sum_{i=2}^{\delta}|T_i|\}$ (see Definition 4).

Lemma 4. *If $M = 0$ then the makespan of the TREE–scheduling algorithm is*

$$\sum_{\substack{u \in V \\ d(s,u) \leq m}} w(u) d(s,u) + m \sum_{i=1}^{\delta}(|B_i| + |C_i|).$$

Proof. By (4), $M = 0$ corresponds to $|B_1|+|C_1| \leq \sum_{i=2}^{\delta} |T_i| - \max\{0, |C_1| - \sum_{i=2}^{\delta}(|T_i| - w(s_i))\}$. We show that whenever a packet is transmitted by s at time $t \geq 1$ to a node u then the successive call by s is scheduled at time

$$t' = t + \begin{cases} m & \text{if } d(s,u) > m, \\ d(s,u) & \text{if } d(s,u) \leq m. \end{cases} \tag{7}$$

This, by Lemma 3, gives the desired result.
Recall that at the beginning of the algorithm TREE-scheduling we set $a_k = |A_k|$, $b_k = |B_k|$, $c_k = |C_k|$, for $k = 1, \ldots, \delta$, and that these values are updated during the execution of the algorithm. Moreover, we denote by $\gamma = |D|$, the cardinality of D, throughout the execution of the algorithm TREE-scheduling; at the beginning of the algorithm we simply have $\gamma = |D| = \delta$. Finally, we denote by $\int_k$ the current number of packets, according to the execution of the algorithm, still to be sent to s_k; initially $\int_k = w(s_k)$, for $k = 1, \ldots, \delta$.

Assume first that $\sum_{k=1}^{\gamma}(b_k + c_k) = 0$, that is, all the remaining packets are to be transmitted to nodes at level at most m. If the time counter τ is t and s is scheduled to transmit to a node u at time t, then τ is incremented by $d(s,u)$. This proves (7) in this case. This applies in particular when $|B_1| + |C_1| = 0$.

We show now that whenever $\sum_{k=1}^{\gamma}(b_k + c_k) > 0$ then $\gamma \geq 2$ and (7) holds.
If (when the algorithm starts) $|B_1| + |C_1| \geq 1$, by hypothesis we have that $\sum_{i=2}^{\gamma} |T_i| \geq |B_1| + |C_1| \geq 1$ and therefore $\gamma \geq 2$; furthermore, the first packet is transmitted at time $\tau = 1$.
Consider any odd-numbered iteration step ℓ of Phase 2 of the algorithm. Let $T_{\ell_1}, \ldots, T_{\ell_\gamma}$ be an ordering of the subtrees such that $b_{\ell_1} + c_{\ell_1} \geq b_{\ell_2} + c_{\ell_2} \geq \cdots \geq b_{\ell_\gamma} + c_{\ell_\gamma}$ at the beginning of step ℓ (recall that $b_{\ell_j} + c_{\ell_j}$ and n_{ℓ_j} refer to the updated number of transmissions still to be done to T_{ℓ_j}).
Assume that the message sent at step $\ell - 1$ was not sent to T_{ℓ_1} and that the condition

$$b_{\ell_1} + c_{\ell_1} \leq \sum_{j=2}^{\gamma} n_{\ell_j} - \max\{0, c_{\ell_1} - \sum_{j=2}^{\gamma}(n_{\ell_j} - \int_{\ell_j})\} \tag{8}$$

holds at the beginning of step ℓ and that the time at which a packet is transmitted at this step is t.
We show that at the beginning of step $\ell + 2$, t has been updated according to (7) and that condition (8) still holds thus proving $\gamma \geq 2$.
Case 1: At the beginning of step ℓ, it holds $b_{\ell_1} + c_{\ell_1} \geq 1$ and $b_{\ell_2} + c_{\ell_2} \geq 1$.
Let the value of the time counter τ be t at the beginning of the iteration step ℓ. Subphase 2.1 is executed at both iteration steps ℓ and $\ell + 1$. Namely,

- during step ℓ, the BS s transmits to T_{ℓ_1} at time t,
- during step $\ell + 1$, the BS s transmits to T_{ℓ_2} at time $t' = t + m$;

moreover the time counter τ is set to $t' + m$ at the end of step $\ell + 1$. This proves that (7) holds; let us now verify (8).

- Suppose either $\gamma = 2$ or $\gamma \geq 3$ and $b_{\ell_3} + c_{\ell_3} < b_{\ell_1} + c_{\ell_1}$. Let a'_ℓ, b'_ℓ and c'_ℓ be the updated values of a_ℓ, b_ℓ and c_ℓ, then at the end of step $\ell + 1$ it holds

$$b'_{\ell_1} + c'_{\ell_1} = b_{\ell_1} + c_{\ell_1} - 1 \geq b'_j + c'_j, \qquad j \neq \ell_1.$$

  One can see (by cases analysis on the value of c_{ℓ_1}), that both the left and right side terms of inequality (8) have been decremented by 1, hence (8) holds again at the beginning of step $\ell + 2$. Furthermore the message was not sent at step $\ell + 1$ to T_{ℓ_1}.

- Suppose now $\gamma \geq 3$ and $b_{\ell_3} + c_{\ell_3} = b_{\ell_1} + c_{\ell_1}$. This implies that also $b_{\ell_2} + c_{\ell_2} = b_{\ell_1} + c_{\ell_1}$. Then at steps ℓ and $\ell + 1$, the BS s transmits to T_{ℓ_1} and T_{ℓ_2}, respectively. At the end of step $\ell + 1$, the first subtree in the reordering (according to the updated sizes of the subtrees) is T_{ℓ_3} (the message at step $\ell + 1$ was not sent to it). Now we show that at the end of step $\ell + 1$, the updated number of unsatisfied requests satisfies $b'_{\ell_3} + c'_{\ell_3} \leq \sum_{j \neq 3} n'_{\ell_j} - \max\{0, c'_{\ell_3} - \sum_{j \neq 3}(n'_{\ell_j} - f'_{\ell_j})\}$. Indeed, we have

$$\sum_{j \neq 3} n'_{\ell_j} = \sum_{j \neq 3}(a'_{\ell_j} + b'_{\ell_j} + c'_{\ell_j}) \geq \sum_{j \neq 3}(1 + b'_{\ell_j} + c'_{\ell_j}) \geq (\gamma - 1) + b'_{\ell_1} + c'_{\ell_1} \geq 2 + b'_{\ell_1} + c'_{\ell_1} = 1 + b'_{\ell_3} + c'_{\ell_3}$$

  and

$$\sum_{j \neq 3}(n'_{\ell_j} - f'_{\ell_j}) \geq c'_{\ell_1} + b'_{\ell_1} + (a'_{\ell_1} - f'_{\ell_1}) \geq c'_{\ell_1} + b'_{\ell_1} = b'_{\ell_3} + c'_{\ell_3} - 1 \geq c'_{\ell_3} \quad (\text{since } b'_{\ell_3} > 0).$$

  Hence, (8) holds also at the beginning of step $\ell + 2$.

Case 2: At the beginning of step ℓ it holds $b_{\ell_1} + c_{\ell_1} \geq 1$ and $b_{\ell_2} + c_{\ell_2} = \cdots = b_{\ell_\gamma} + c_{\ell_\gamma} = 0$. Let the value of the time counter τ be t at the beginning of the iteration step ℓ.
Consider the step ℓ, the root s transmits to T_{ℓ_1} at time t and the time counter τ is updated to $t + m$. Consider now the step $\ell + 1$. By (8), there exists at least one $\ell_j \neq \ell_1$ with $n_{\ell_j} > 0$. We distinguish three cases.

- $\sum_{j=2}^{\gamma}(n_{\ell_j} - f_{\ell_j}) \geq 1$.
  Step $\ell + 1$: the root s transmits to a node u at level h ($2 \leq h \leq m$), in some T_{ℓ_j}, with $\ell_j \neq \ell_1$, at time $t + m$ and the time counter τ is updated to $t + m + d(s, u)$.

- $n_{\ell_j} = f_{\ell_j}$, for $j = 2, \ldots \gamma$, and $c_{\ell_1} \geq 1$.
  Step $\ell + 1$: the root s transmits at both times $t + m$ and $t + m + 1$ to the roots of 2 subtrees different from T_{ℓ_1} (notice that (8) implies $\sum_{j=2}^{\gamma} n_{\ell_j} \geq 2$) and the time counter τ is updated to $t + m + 2$.

- $n_{\ell_j} = \int_{\ell_j}$, for $j = 2, \dots \gamma$, and $c_{\ell_1} = 0$.
  Step $\ell + 1$: the root s transmits at time $t + m$ to the root of a subtree different from T_{ℓ_1} and the time counter τ is updated to $t + m + 1$.

In each of the above cases (7) holds. It is easy to see that left side and right side of inequality (8) are both decreased by the same quantity, therefore, (8) also holds at the beginning of step $\ell + 2$. □

Lemma 5. *If the input tree T satisfies $M = |B_1| + |C_1| - \sum_{i=2}^{\delta} |T_i| + \max\{0, |C_1| - \sum_{i=2}^{\delta}(|T_i| - w(s_i))\} > 0$ then the makespan is*

$$\sum_{\substack{u \in V \\ d(s,u) \le m}} w(u)\, d(s,u) + m \sum_{i=1}^{\delta}(|B_i| + |C_i|) + |B_1| + |C_1| - \sum_{i=2}^{\delta} |T_i| + \max\{0,\ |C_1| - \sum_{i=2}^{\delta}(|T_i| - w(s_i))\}.$$

Proof. Again, it is useful to recall that at the beginning of the algorithm TREE-scheduling we set $\gamma = |D| = \delta$, $a_k = |A_k|$, $b_k = |B_k|$, $c_k = |C_k|$, $\int_k = w(s_k)$, for $k = 1, \dots, \delta$, and that these values are updated during the execution of the algorithm.

Consider the relation

$$b_1 + c_1 > \sum_{i=2}^{\gamma} n_i - \max\{0, c_1 - \sum_{i=2}^{\gamma}(n_i - \int_i)\}. \tag{9}$$

We first show that as long as $n_j > 0$ for some $j > 1$ then (9) holds at the beginning of any odd-numbered step of Phase 2 in the algorithm, that is, the first subtree in the $\succeq$ ordering is always T_1.

Relation (9) holds by hypothesis at the beginning of the algorithm, that is, at step 0.
Consider any two consecutive steps, say ℓ and $\ell + 1$ with ℓ odd. Let the value of the time counter τ be t at the beginning of step ℓ.
During step ℓ the root s transmits to T_1 at time t and the time counter is updated to $t + m$. At the beginning of step $\ell + 1$, one of the following cases can occur:

1) $\sum_{i=2}^{\gamma}(n_i - \int_i) \ge 1$. The root s transmits to a node u in some T_i with $i > 1$ at time $t + m$; the time counter is incremented by $\min\{d(s,u), m\}$.

2) $n_{\ell_j} = \int_{\ell_j}$, for $j = 2, \dots \gamma$ and $c_1 \ge 1$. The root s transmits either twice to nodes at level 1 (in subtrees different from T_1) at both times $t + m$ and $t + m + 1$ or it transmits only at time $t + m$ to the single unsatisfied node (at level 1) remaining in a subtree different from T_1; the time counter is incremented by 2.

3) $n_{\ell_j} = \int_{\ell_j}$, for $j = 2, \dots \gamma$ and $c_1 = 0$. The root s transmits to a node at level 1 in a subtree different from T_1 and the time counter is incremented by 1.

In each of the above cases the decrement on the right and left terms of the inequality (9) during the execution of steps ℓ and $\ell+1$ implies that (9) also holds at the beginning of step $\ell+2$ (or $n_j = 0$ for $j > 1$).

We now determine the makespan. For that we compute the increment of the counter in phase 2, that is the value of the time counter after s has transmitted the last packet destined to a node in some T_j with $j \neq 1$. We notice that case 1) occurs $p_1 = \sum_{i=2}^{\delta}(|T_i| - w(s_i))$ times, therefore, during the corresponding pairs of steps (ℓ and $\ell+1$ in the above notation), the value of τ is incremented by

$$mp_1 \quad + \quad \sum_{i=2}^{\delta} \sum_{\substack{u \in V(T_i) \\ d(s,u) \geq 2}} \min\{d(s,u),\, m\} = mp_1 + m\sum_{i=2}^{\delta}(|B_i| + |C_i|) + \sum_{\substack{u \notin V(T_1) \\ 2 \leq d(s,u) \leq m}} w(u)\, d(s,u).$$

Then we note that if case 2) appears p_2 times the time counter is increased by $(m+2)p_2$ and if case 3) appears p_3 times the time counter is increased by $(m+1)p_3$. During phase 3, only packets destined to nodes of T_1 remain to be transmitted one after the other. If the remaining n_1' packets to be transmitted to nodes in T_1 are subdivided into $n_1' = c_1' + b_1' + a_1'$, the increment of the time counter is $m+2$ for each of the c_1' packets transmitted to the nodes at level $m+2$ or more, $m+1$ for each of the b_1' packets transmitted to nodes at level $m+1$, and $d(s,u)$ for each u among the a_1' nodes at level m or less. As it is always the case that $a_1' = a_1 = |A_1|$, the increase of the counter will be

$$(m+2)p_2 + (m+1)p_3 + (m+2)c_1' + (m+1)b_1' + \sum_{\substack{u \in V(T_1) \\ d(s,u) \leq m}} w(u)\, d(s,u).$$

As

$$\sum_{\substack{u \notin V(T_1) \\ 2 \leq d(s,u) \leq m}} w(u)\, d(s,u) + \sum_{i=2}^{\delta} w(s_i) + \sum_{\substack{u \in V(T_1) \\ d(s,u) \leq m}} w(u)\, d(s,u) = \sum_{\substack{u \in V \\ d(s,u) \leq m}} w(u) d(s,u)$$

we get that the last transmission (of the packet destined to the root of T_1) is scheduled at time

$$\tau = \sum_{\substack{u \in V \\ d(s,u) \leq m}} w(u)\, d(s,u) + m\sum_{i=2}^{\delta}(|B_i| + |C_i|) - \sum_{i=2}^{\delta} w(s_i) + mp_1 + (m+2)p_2 + (m+1)p_3 + (m+2)c_1' + (m+1)b_1'. \tag{10}$$

To evaluate the values of p_2, p_3, c_1', b_1' we will distinguish three cases.

a) $|C_1| \leq p_1 = \sum_{i=2}^{\delta}(|T_i| - w(s_i))$: Under this hypothesis case 2) never occurs so $p_2 = 0$, while case 3) occurs $p_3 = \sum_{i=2}^{\delta} w(s_i)$ times, and $c_1' = 0$, $\; b_1' = |B_1| - (p_1 - |C_1|) - \sum_{i=2}^{\delta} w(s_i) =$

$|B_1| + |C_1| - \sum_{i=2}^{\delta} |T_i|$. Therefore, by (10) we get

$$\tau = \sum_{\substack{u \in V \\ d(s,u) \leq m}} w(u)\, d(s,u) + m \sum_{i=1}^{\delta} (|B_i| + |C_i|) + |B_1| + |C_1| - \sum_{i=2}^{\delta} |T_i|.$$

b) $|C_1| > p_1$ and $\sum_{i=2}^{\delta} w(s_i) \leq 2(|C_1| - p_1)$: Under this hypothesis case 2) occurs $p_2 = \left\lceil \sum_{i=2}^{\delta} w(s_i)/2 \right\rceil$ times, but case 3) does not occur; $c'_1 = |C_1| - p_1 - p_2$, $b'_1 = |B_1|$. Therefore, by (10) we get

$$\tau = \sum_{\substack{u \in V \\ d(s,u) \leq m}} w(u)\, d(s,u) + m \sum_{i=1}^{\delta} (|B_i| + |C_i|) + |B_1| + 2|C_1| - 2 \sum_{i=2}^{\delta} |T_i| + \sum_{i=2}^{\delta} w(s_i).$$

c) $|C_1| > p_1$ and $\sum_{i=2}^{\delta} w(s_i) > 2(|C_1| - p_1)$. Under this hypothesis case 2) occurs $p_2 = |C_1| - p_1$ times and case 3) occurs $p_3 = \sum_{i=2}^{\delta} w(s_i) - 2p_2$ times. Furthermore $c'_1 = 0$, $b'_1 = |B_1| - p_3$. By (10), we get again the same value of τ as in case b).

□

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 need to receive packets from the BS s, that is, we assume $w(v) \geq 0$, for each $v \in V - \{s\}$.

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 $T_1, T_2, \cdots, T_\delta$ the subtrees of T rooted at the children of s. We present an algorithm which gives an optimal schedule in time $O(\delta W^{4\delta})$, where $W = \sum_{v \in V - \{s\}} w(v)$ is the number of items to be transmitted (i.e, the sum of the node weights).

Lemma 6. *Consider $u, v \in V$. If $u, v \in V(T_i)$, for some $1 \leq i \leq \delta$, and either of the following conditions holds*

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

b) $d(s,u) \geq d(s,v) \geq m+2$

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

Proof. Let S be a schedule where s transmits to node v before than to node u. Consider a new schedule S' where s transmits in the same order as in S except for the transmissions to v and u that are exchanged. We show that $\mathcal{T}_S \geq \mathcal{T}_{S'}$, where $\mathcal{T}_S$ and $\mathcal{T}_{S'}$ represent the makespan of S and S', respectively (cfr (1)).

First let u and v be such that $2 \leq d(s,u) \leq d(s,v) \leq m$. Consider the schedule S.

– For any x to which s transmits before v in S, we trivially have $\mathcal{T}_{S'}(x) = \mathcal{T}_S(x)$.

– Consider now a node x to which s transmits after u in S: since we suppose $d(s,u) \geq 2$, the transmission following u cannot interfere with the one preceding u (cfr. Fact 1) and the order of transmission to u and v is not relevant for x. Therefore, also in this case we have $\mathcal{T}_{S'}(x) = \mathcal{T}_S(x)$.

– If s transmits to x after v but before u (excluding $x = v$) in S then $\mathcal{T}_{S'}(x)$ takes into account the time $d(s,u)$ spent to transmit to u (cfr. Fact 1) instead of $d(s,v)$ as in S; therefore we have $\mathcal{T}_{S'}(x) = \mathcal{T}_S(x) - (d(s,v) - d(s,u)) \leq \mathcal{T}_S(x)$.

– Furthermore, we have $\mathcal{T}_{S'}(u) = \mathcal{T}_S(v)$, and so $d(s,u) + \mathcal{T}_{S'}(u) - 1 \leq d(s,v) + \mathcal{T}_S(v) - 1$ $\mathcal{T}_{S'}(v) = \mathcal{T}_S(u) + d(s,u) - d(s,v)$ and so $d(s,v) + \mathcal{T}_{S'}(v) - 1 = d(s,u) + \mathcal{T}_S(u) - 1$.

  In conclusion,

$$\mathcal{T}_{S'} = \max_{\substack{x \in V \\ w(x)>0}} \{d(s,x) + \mathcal{T}_{S'}(x) - 1\ \} \leq \max_{\substack{x \in V \\ w(x)>0}} \{d(s,x) + \mathcal{T}_S(x) - 1\ \} = \mathcal{T}_S. \qquad (11)$$

Consider now the case $d(s,u) \geq d(s,v) \geq m+2$ and $u, v \in V(T_i)$.
Under this hypothesis, by Fact 1, we immediately get

$$\begin{aligned}
&d(s,x) + \mathcal{T}_{S'}(x) - 1 = d(s,x) + \mathcal{T}_S(x) - 1 && \text{for any } x \neq u, v \\
&d(s,u) + \mathcal{T}_{S'}(u) - 1 = d(s,u) + \mathcal{T}_S(v) - 1 \leq d(s,u) + \mathcal{T}_S(u) - 1 && \\
&d(s,v) + \mathcal{T}_{S'}(v) - 1 = d(s,v) + \mathcal{T}_S(u) - 1 \leq d(s,u) + \mathcal{T}_S(u) - 1 &&
\end{aligned}$$

Therefore, as in (11), we have $\mathcal{T}_{S'} \leq \mathcal{T}_S$. □

We want to compute a shortest schedule with relative ordering given by Lemma 6 using dynamic programming. To this aim we consider the following lists C_i, B_i, and A_i for $i = 1, \ldots, \delta$, where

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

- $B_i = (z_{i,1}, z_{i,2}, \cdots)$ 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 each node $z_{i,j}$ being repeated $w(z_{i,j})$ times.

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

Notice that in each subtree, the nodes at a same level can be ordered in an arbitrary way.
Given integers $c_i \leq |C_i|$, $b_i \leq |B_i|$, $a_i \leq |A_i|$, $r_i \leq |w(s_i)|$, for $i = 1, \ldots \delta$, we denote by

$$S(c_1, \ldots, c_\delta, b_1, \ldots, b_\delta, a_1, \ldots, a_\delta, r_1, \ldots, r_\delta), \qquad (12)$$

an optimal schedule satisfying Lemma 6 when for each subtree T_i, for $i = 1, \ldots, \delta$, the only packets to be transmitted are those destined to:

- the first c_i nodes of C_i,
- the first b_i nodes of B_i,
- the first a_i nodes of A_i, and
- r_i times the root s_i of T_i.

In the following we will use the compact vectorial notation

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

Therefore, we write $S(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r})$ for $S(c_1, \cdots, c_\delta, b_1, \cdots, b_\delta, a_1, \cdots, a_\delta, r_1, \cdots, r_\delta)$. Moreover, let

$$S(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, \mathit{type}))$$

be an optimal schedule for (12) with the additional restriction that the first transmission in the schedule is to a node in T_j where $\mathit{type} \in \{r, C, B, A\}$ specifies whether this node is either the root of T_j, or a node in C_j (by Lemma 6, node x_{j,c_j}), or a node in B_j, or in A_j (by Lemma 6, node y_{j,a_j}).

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

$$\mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}) = \min_{1 \le j \le \delta} \min_{\mathit{type} \in \{r, C, B, A\}} \mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, \mathit{type})). \tag{13}$$

Denote by $\mathbf{e}_i$ the identity vector $\mathbf{e}_i = (e_{i,1}, \cdots, e_{i,\delta})$ 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 $j = 1, \cdots, \delta$, it holds*

- *if $\mathit{type} = r$, i.e., the first transmission is for the root s_j of T_j, 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 $\mathit{type} = A$, i.e., the first transmission is for $y_{j,a_j} \in A_j$ 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 $\mathit{type} = B$, i.e., the first transmission is for $z_{j,b_j}, \in B_j$, then*

$$\mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, B)) = \min_k \min_{\mathit{type}'} \begin{cases} m + \max\{1,\ \mathcal{T}(\mathbf{c}, \mathbf{b} - \mathbf{e}_j, \mathbf{a}, \mathbf{r}, (k, \mathit{type}'))\} & \text{if } j \neq k \\ \\ m + 1 + \mathcal{T}(\mathbf{c}, \mathbf{b} - \mathbf{e}_j, \mathbf{a}, \mathbf{r}, (k, \mathit{type}')) & \text{if } j = k \end{cases}$$

- *if $type = C$, i.e., the first transmission is for $x_{j,c_j}, \in C_j$, then*

  $\mathcal{T}(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, C)) =$

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

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 $S(\mathbf{c}, \mathbf{b}, \mathbf{a}, \mathbf{r}, (j, type))$; starting from $S(\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (j, type))$ whose makespan is $\mathcal{T}(\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (j, type)) = 0$, for each j and $type$, where $\mathbf{0} = (0, \ldots, 0)$ is the null vector.

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

Theorem 2. *It is possible to obtain an optimal schedule in time $O(\delta W^{4\delta})$.*

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.

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-SPscheduling (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 $\ell = 1, \ldots, h$
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 7. *The makespan of the scheduling produced by Graph-SPscheduling(G, SP, s) is*

$$\max\left\{\sum_{\substack{v\in V\\ d(s,v)\le m+1}} w(v)d(s,v) + (m+2)\sum_{\substack{v\in V\\ d(s,v)\ge m+2}} w(v), \quad \max_{\ell\ge m+2}\left\{\ell - m - 2 + (m+2)\sum_{\substack{v\in V\\ d(s,v)\ge \ell}} w(v)\right\}\right\}. \tag{14}$$

Proof. According to the Graph-SPscheduling algorithm, the first packet is sent at time 1. For any ℓ, with $1 \le \ell \le h$, the last of the packets to nodes at distance ℓ or more from s is transmitted at time

$$1 - \min\{\ell, m+2\} + \sum_{\substack{v\in V\\ d(s,v)\ge \ell}} w(v)\min\{d(s,v), m+2\}.$$

Since $\ell - 1$ more time slots are necessary for this last packet to reach its destination, we have that the largest time at which a node at distance ℓ or more from s receives its packet is

$$\mathcal{T}_\ell = \ell - \min\{\ell, m+2\} + \sum_{\substack{v\in V\\ d(s,v)\ge \ell}} w(v)\min\{d(s,v), m+2\}. \tag{15}$$

The makespan of Graph-SPscheduling, is $\max_{1\le\ell\le h} \mathcal{T}_\ell$, which gives the desired result by noticing that

$$\mathcal{T}_\ell = \begin{cases} \sum_{\substack{v\in V\\ \ell\le d(s,v)\le m+1}} w(v)d(s,v) + (m+2)\sum_{\substack{v\in V\\ d(s,v)\ge m+2}} w(v) & \text{if } \ell \le m+1 \\ \\ \ell - m - 2 + (m+2)\sum_{\substack{v\in V\\ d(s,v)\ge \ell}} w(v) & \text{if } \ell \ge m+2 \end{cases}$$

□

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 packet transmissions on a (shortest path) tree, we *still* need to deal with possible collisions due to the edges in $E - E(SP)$.

Lemma 8. *There is no interference between any two calls done by the GRAPH-scheduling algorithm.*

Proof. In order to see that our algorithm does not suffer from interferences, let us first notice that if $(x, y) \in E$ then $|d(s, x) - d(s, y)| \leq 1$, that is, the levels of nodes x and y in SP differ by at most 1. Moreover, if s transmits to u at time t and then to v at time $t' > t$ then the Graph-SPscheduling algorithm imposes that $t' = t+\min\{d(s, u),\ m+2\}$. Furthermore, the path followed by the packet for u (resp. for v) is the path from s to u (resp. for v) in the shortest path tree T.
This implies that, if $d(s, u) \leq m + 2$, then the packet for u arrives before that the packet for v leaves s; hence, no interference can occur.
Let $d(s, u) > m+2$. At time $t' = t+m+2+h$, the packet for u is either arrived or transmitted from a node x at level $m+2+h$ to a node y at level $m+3+h$ and the packet for v is transmitted from a node x' at level h to a node y' at level $h+1$ both in the shortest path routing tree T. As $d(s, x) \leq d(s, x') + d(x, x')$ we get that $d(x, x') \geq m + 2$ and so there is no interference between the two calls done at time t'. □

We derive now the approximation factor of the algorithm.

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-SPscheduling(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 schedule for G.

Proof. The following lower bound on the makespan of any schedule was introduced in [1].

$$\mathcal{T}^*(G) \geq \max\left\{ \sum_{\substack{v\in V\\ d(s,v)\leq m}} w(v)d(s, v) + m \sum_{\substack{v\in V\\ d(s,v)\geq m+1}} w(v),\ \max_{\ell\geq m+1} \left\{ \ell - m + m \sum_{\substack{v\in V\\ d(s,v)\geq \ell}} w(v) \right\}\right\}. \quad (16)$$

We bound $\frac{\mathcal{T}}{\mathcal{T}^*(G)}$ by evaluating the ratio of the makespan of (14) to (16).

To this aim, we notice that for any positive integers a, b, c, d it holds $\frac{\max\{a,c\}}{\max\{b,d\}} \leq \max\left\{\frac{a}{b}, \frac{c}{d}\right\}$; hence, we can separately bound the two ratios

$$R_1 = \frac{\sum_{\substack{v\in V\\ d(s,v)\leq m+1}} w(v)d(s, v) + (m + 2) \sum_{\substack{v\in V\\ d(s,v)\geq m+2}} w(v)}{\sum_{\substack{v\in V\\ d(s,v)\leq m}} w(v)d(s, v) + m \sum_{\substack{v\in V\\ d(s,v)\geq m+1}} w(v)}$$

and

$$R_2 = \frac{\max_{\ell \geq m+2} \left\{ \ell - m - 2 + (m+2) \sum_{\substack{v \in V \\ d(s,v) \geq \ell}} w(v) \right\}}{\max_{\ell \geq m+1} \left\{ \ell - m + m \sum_{\substack{v \in V \\ d(s,v) \geq \ell}} w(v) \right\}}.$$

and show that both of them are upper bounded by the desired value $1 + \frac{2}{m}$.

We have

$$\begin{aligned}
R_1 &\leq \frac{\sum_{\substack{v \in V \\ d(s,v) \leq m}} w(v)d(s,v) + (m+2) \sum_{\substack{v \in V \\ d(s,v) \geq m+1}} w(v)}{\sum_{\substack{v \in V \\ d(s,v) \leq m}} w(v)d(s,v) + m \sum_{\substack{v \in V \\ d(s,v) \geq m+1}} w(v)} \\
&= 1 + \frac{2 \sum_{\substack{v \in V \\ d(s,v) \geq m+1}} w(v)}{\sum_{\substack{v \in V \\ d(s,v) \leq m}} w(v)d(s,v) + m \sum_{\substack{v \in V \\ d(s,v) \geq m+1}} w(v)} \leq 1 + \frac{2}{m}
\end{aligned}$$

where the last inequality follows since $\sum_{\substack{v \in V \\ d(s,v) \leq m}} w(v)d(s,v) \geq 0$.
Moreover,

$$\begin{aligned}
R_2 &= \frac{\max_{\ell \geq m+2} \left\{ \ell - m - 2 + (m+2) \sum_{\substack{v \in V \\ d(s,v) \geq \ell}} w(v) \right\}}{\max_{\ell \geq m+1} \left\{ \ell - m + m \sum_{\substack{v \in V \\ d(s,v) \geq \ell}} w(v) \right\}} \leq \frac{\max_{\ell \geq m+1} \left\{ \ell - m + (m+2) \sum_{\substack{v \in V \\ d(s,v) \geq \ell}} w(v) \right\}}{\max_{\ell \geq m+1} \left\{ \ell - m + m \sum_{\substack{v \in V \\ d(s,v) \geq \ell}} w(v) \right\}} \\
&\leq \max_{\ell \geq m+1} \left\{ \frac{\ell - m + (m+2) \sum_{\substack{v \in V \\ d(s,v) \geq \ell}} w(v)}{\ell - m + m \sum_{\substack{v \in V \\ d(s,v) \geq \ell}} w(v)} \right\} \qquad \text{(By iteratively applying } \tfrac{\max\{a,c\}}{\max\{b,d\}} \leq \max\left\{\tfrac{a}{b}, \tfrac{c}{d}\right\}\text{)} \\
&= \max_{\ell \geq m+1} \left\{ 1 + \frac{2 \sum_{\substack{v \in V \\ d(s,v) \geq \ell}} w(v)}{\ell - m + m \sum_{\substack{v \in V \\ d(s,v) \geq \ell}} w(v)} \right\} \leq \max_{\ell \geq m+1} \left\{ 1 + \frac{2}{m} \right\} = 1 + \frac{2}{m}
\end{aligned}$$

□

4 Complexity Results

In this section we 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 our problem.

MTDG (Minimum Time Data Gathering)
Instance: A graph $G = (V, E)$, an interference range $m \geq 2$, a BS $s \in V$, 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 transmission schedule on it of the $w(v)$ packets sensed at v, for each $v \in V$, to the base station s so that the whole process is collision–free, and the makespan is $\mathcal{T} \leq K$?

As in the previous sections, we actually consider the equivalent diffusion problem

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 the NP-hardness of **MTPB** by a reduction from the well known Partition Problem [13].

PARTITION
Instance: $n+1$ integers $a_1, a_2, \cdots, a_n, B$ such that $\sum_{i=1}^{n} a_i = 2B$.
Question: Is there a subset $S \subset \{1, 2, \cdots, n\}$ such that $\sum_{i \in S} a_i = B$?

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

- The graph (c.f.r. Figure 4) is $G = (V, E)$ with node set

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

  edge set

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

  and node weights

$$\begin{array}{ll}
w(u_j^0) = w(v_j^0) = 0, & \text{for } j = 1\ldots, m+1, \\
w(u_j^0) = w(v_j^0) = 1, & \text{for } j = m+2\ldots, m+n+1, \\
w(u_j^i) = w(v_j^i) = 0, & \text{for } i = 1\ldots, n \text{ and } j = 0\ldots, m, \\
w(x^i) = a_i, & \text{for } i = 1\ldots, n.
\end{array}$$

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

The structure of the MTPB instance is shown in Figure 4. We notice that the graph G can be constructed in polynomial-time. We show now 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 $\mathcal{T} \leq K$.

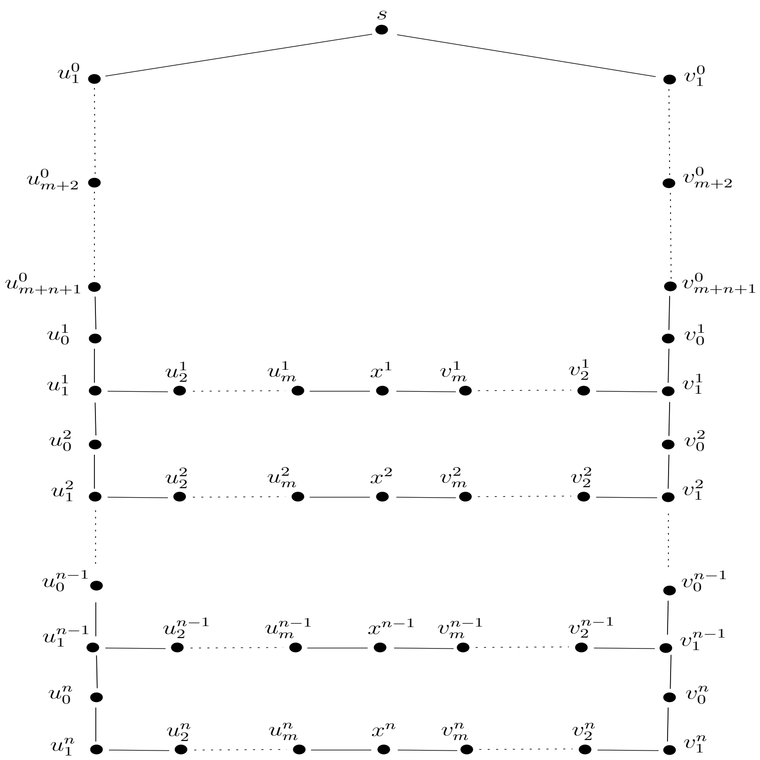


Figure 4. MTPB instance associated to a PARTITION instance.

Lemma 9. *If the* **PARTITION** *instance has a solution S such that $\sum_{i\in S} a_i = B$ then the makespan is $\mathcal{T} \leq K$ where $K = 2m(B+n)+2$.*

Proof. Denote by $\overline{S} = \{1, 2, \cdots, n\} - S$ the complement of S. Consider the tree $T = (V(T), E(T))$ of G rooted at s which consists of all the edges of G which belong to:

(1) the paths $(s, u_1^0, \ldots, u_{m+n+1}^0, u_0^1, u_1^1, \ldots, u_0^i, u_1^i, u_2^i, \ldots, u_m^i, x^i)$ in G from s to x^i via u_1^0, for each $i \in S$,

(2) the paths $(s, v_1^0, \ldots, v_{m+n+1}^0, v_0^1, v_1^1, \ldots, v_0^j, v_1^j, v_2^j, \ldots, v_m^j, x^j)$ in G from s to x^j via v_1^0, for each $j \in \overline{S}$.

Notice that even though T is not a spanning tree of G, it spans all the nodes of non–zero weight in G.

Call T_1 the subtree of T rooted at u_1^0, and T_2 the subtree of T rooted at v_1^0. The hypothesis $\sum_{i \in S} a_i = B = \sum_{i \in \overline{S}} a_i$ implies that $B + n$ packets are to be transmitted by s both in T_1 and in T_2. Consider now the following schedule of the packets in T:

- alternately, s transmits a packet to T_1 and one to T_2,
  (s transmits no packet to a node at level ℓ in T_i before transmitting all the packets to nodes at level $\ell + 1$ or more in T_i, $i = 1, 2$).

- if s transmits a packet at time t then the next packet is sent at time $t + m$.

In the above schedule, recalling that $w(x^i) = a_i$, for $1 \leq i \leq n$, and $d(s, x^i) = 2m + n + 2i + 1 \leq 2m + 3n + 1$, we get that s transmits all packets of nodes x^i by time $1 + m(2B - 1)$; each of these packets reaches its destination by time $m(2B - 1) + 2m + 3n + 1 \leq K$ (recall that $m \geq 2$). Furthermore, since $w(u_j^0) = w(v_j^0) = 1$, for $m+2 \leq j \leq m+n+1$, and $d(s, u_{m+2}^0) = d(s, v_{m+2}^0) = m + 2$, we get that s transmits to all these nodes by time $(2B + 2n - 1)m + 1$; each of these packets reaches its destination by time $2m(B + n) + 2 = K$.
Finally, we have only to prove that no interference occurs during the above scheduling. We first observe that two nodes in different subtrees of T are connected in G also by a path not passing through s, however such a path contains at least $m - 1$ nodes in $V - V(T)$ (which do not participate to the process). Indeed, an internal node (i.e., a transmitting node) in a subtree of T has a distance at least m from any node which belongs to the other subtree of T. Using this and Fact 1, we know that for any two nodes u and v the calls done during the transmission from s to u and the calls of the transmission from s to v never interfere. $\square$

Lemma 10. *If there is a schedule for the* **MTPB** *instance with makespan $\mathcal{T} \leq K$ then the* **PARTITION** *instance has a solution S such that $\sum_{i \in S} a_i = B$.*

Proof. Suppose that there is a schedule for the **MTPB** instance such that the makespan is $\mathcal{T} \leq K$. We are ready to show that the PARTITION instance has a solution. Any schedule for the **MTPB** instance gives a path, say $P(x^i)$, from s to x^i, for each $1 \leq i \leq n$, since x^i has at least one packet to receive from s and the assumption that the routing is performed on a tree implies that all the a_i packets destined to x^i go through the same path. Furthermore, the use of a routing tree implies that the paths of the packets destined to nodes which lay on $P(x^i)$ are fixed (to the corresponding subpath of $P(x^i)$). Define now

$$S = \{i \mid s \text{ transmits the } a_i \text{ packets of } x^i \text{ through } u_1^0\},$$

$$\overline{S} = \{j \mid s \text{ transmits the } a_j \text{ packets of } x^j \text{ through } v_1^0\}.$$

We claim that S is a solution to the **PARTITION** instance. Assume by contradiction that $\sum_{i \in S} a_i \neq \sum_{j \in \overline{S}} a_j$. Without loss of generality, let $\sum_{i \in S} a_i \geq \sum_{j \in \overline{S}} a_j + 2$ (note that $\sum_{i=1}^n a_i$ is even). It is obvious that the tree used by the transmissions of the packets from s has two subtrees:

the one rooted at u_1^0 whose nodes have to receive $\sum_{i\in S} a_i + n$ packets and the one rooted at v_1^0 whose nodes have to receive $\sum_{j\in \overline{S}} a_j + n$ packets. Hence, there are at least $(\sum_{i\in S} a_i + n) - (\sum_{j\in \overline{S}} a_j + n) \geq 2$ packets that need to be sent successively by s to nodes of a same subtree. Considering that all the nodes with positive weight are at distance $\geq m+2$ from s, by Fact 1 we have that if s transmits at time t a packet to a node in T_1 (resp. T_2) then (1) s cannot transmit another packet to some node in T_2 (resp. T_1) before time $t+m$. (2) s cannot transmit another packet to some node in T_1 (resp. T_2) before time $t+m+2$. Finally, the last packet sent by s needs at least $m+2$ time slots in order to reach its destination, since the nodes with positive weight are at distance at least $m+2$ from s. In conclusion, we have

$$\begin{aligned}\mathcal{T}(G) &\geq m\left(\sum_{i\in S} a_i + \sum_{j\in \overline{S}} a_j + 2n - 1\right) + 2\left(\sum_{i\in S} a_i - \sum_{j\in \overline{S}} a_j - 1\right) + (m+2)\\ &\geq m\left(\sum_{i=1}^{n} a_i + 2n\right) + 2 + 2 > K\end{aligned}$$

□

Hence we get

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

5 Open Problems

Several interesting questions remain open.

- Determine the complexity of gathering in trees, with arbitrary weight distribution, in the case the degree of the root is a function of the number of nodes in the network.
  Indeed, the proposed $O(\delta W^{4\delta})$ algorithm leaves open the problem of the existence of a polynomial algorithm when the BS degree δ depends on the size of the tree even if $w(v) \in \{0,1\}$, for each node v in the tree.
- Determine the complexity of gathering in general graphs, without any restriction on the kind of routing used.
- Determine the complexity of gathering in general graphs, for $m = 1$, if the process must be performed along the edges of a routing tree.
- Determine the complexity of gathering in general graphs in case that all weights are 1, for $m \geq 2$. Indeed, our NP–hardness result holds only in the case some weights can be 0.

References

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

[2] J-C. Bermond, L. Gargano, A. Rescigno, "Gathering with minimum completion time in sensor tree networks", *Journal of Interconnection Networks*, **11** (1-2), (2010) 1–33.

[3] J-C. Bermond, R. Correa, M.-L. Yu, "Optimal Gathering Protocols on Paths under Interference Constraints", *Discrete Mathematics*, **309**(18), (2009), 5574–5587.

[4] J-C. Bermond, J. Peters, "Efficient gathering in radio grids with interference", *Proc. AlgoTel'05, Presqu'île de Giens*, (2005) 103–106.

[5] J-C. Bermond, M.-L. Yu, "Optimal gathering algorithms in multi-hop radio tree networks with interferences", *Ad Hoc and Sensor Wireless Networks*, **9**(1–2), (2010), 109–128.

[6] P. Bertin, J-F. Bresse, B. Le Sage. Accès haut débit en zone rurale: une solution "ad hoc". France Telecom R&D, 22:16–18, 2005.

[7] V. Bonifaci, P. Korteweg, A. Marchetti-Spaccamela, L. Stougie, "An Approximation Algorithm for the Wireless Gathering Problem", *Operations Research Letters* **36** (5), (2008) 605–608.

[8] V. Bonifaci, R. Klasing, P. Korteweg, A. Marchetti-Spaccamela, L. Stougie, "Data Gathering in Wireless Networks", *Graphs and Algorithms in Communication Networks*, A.Koster and X. Munoz editors, Springer Monograph, (2010) 357–377.

[9] H. Choi, J. Wang, E.A. Hughes, "Scheduling for information gathering on sensor network", *Wireless Network*, **15**, (2009) 127–140.

[10] C. Florens, M. Franceschetti, R.J. McEliece, "Lower Bounds on Data Collection Time in Sensory Networks", *IEEE J. on Selected Areas in Communication* **22** (6), (2004) 1110–1120.

[11] L. Gargano, "Time Optimal Gathering in Sensor Networks", *Proc. SIROCCO 2007* (LNCS 4474), (2007) 7-10.

[12] L. Gargano, A.A. Rescigno, "Optimally Fast Data Gathering in Sensor Networks", *Discrete Applied Mathematics* **157**, (2009) 1858–1872.

[13] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman (1979)

[14] L. Gasieniec, I Potapov, "Gossiping with Unit Messages in Known Radio Networks", *IFIP TCS*, (2002) 193–205.

[15] S. Schmid, R. Wattenhofer, "Algorithmic models for sensor networks", *IPDPS 2006*, (2006).

[16] X. Zhu, B. Tang, H. Gupta, "Delay efficient data gathering in sensor networks", *Proc. of MSN 2005* (LNCS 3794), (2005) 380–389.