

# Lower Bounds on Data Collection Time in Sensory Networks

Cédric Florens, *Student Member, IEEE*, Massimo Franceschetti, *Member, IEEE*, and Robert J. McEliece, *Fellow, IEEE*

**Abstract**—Data collection, i.e., the aggregation at the user location of information gathered by sensor nodes, is a fundamental function of sensory networks. Indeed, most sensor network applications rely on data collection capabilities, and consequently, an inefficient data collection process may adversely affect the performance of the network. In this paper, we study via simple discrete mathematical models, the time performance of the data collection and data distribution tasks in sensory networks. Specifically, we derive the minimum delay in collecting sensor data for networks of various topologies such as line, multiline, and tree and give corresponding optimal scheduling strategies. Furthermore, we bound the data collection time on general graph networks. Our analyses apply to networks equipped with directional or omnidirectional antennas and simple comparative results of the two systems are presented.

**Index Terms**—Data collection, delay, sensory networks.

## I. INTRODUCTION

THE advancement of very large scale integration (VLSI) technology has contributed much to the development of microsensor systems. Such systems can combine signal processing, data storage, wireless communication capabilities, and energy sources on a single chip. Possibly distributed over a wide area, it is envisioned that, networks of such devices autonomously perform various sensing tasks such as environmental (seismic, meteorological) monitoring and military surveillance [1]. These networks are referred to as wireless ad hoc sensor networks or simply sensor/sensory networks/webs. In sensor networks, while each node may be mobile, it is typically the case that once the target site of their sensing application is reached a semipermanent stationary configuration is adopted for the purpose of gathering information. In most applications, raw data accumulated at sensor nodes is preprocessed (at this stage, redundancy in the data may be removed) and forwarded to some processing center (end-user) to be analyzed. The processing center will be referred to as the base station (BS) of the sensor network in the rest of this paper.

Sensor networks belong to a larger class of wireless networks called ad hoc networks in the sense that they are normally deprived of an infrastructure unlike traditional cellular networks.

Manuscript received July 15, 2003; revised February 1, 2004. This work was supported in part by the Caltech Lee Center for Advanced Networking.

C. Florens and R. J. McEliece are with the Electrical Engineering Department, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: florens@systems.caltech.edu; rjm@systems.caltech.edu).

M. Franceschetti is with the Electrical Engineering and Computer Sciences Department, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: massimof@eecs.berkeley.edu).

Digital Object Identifier 10.1109/JSAC.2004.830927

They are characterized by an all-to-one communication paradigm, which corresponds to the retrieval of sensor data by the BS. In the area of general ad hoc networks, as well as sensor webs, research has focused on routing [2], medium access control (MAC) [3], [4], and physical layer [5]. References [6] and [7] are protocol suites specifically designed for sensor webs. Furthermore, theoretical results regarding capacity of general static ad hoc networks first appeared in [8]. Finally, most relevant to our research is the so-called packet routing problem, which consists of moving packets of data from one location to another as quickly as possible in a network and has been studied in conjunction with wireline and wireless network models (see, for example, [9]–[12]).

In this paper, we derive new results specific to sensor networks, where in particular, nonuniform data distribution over the network is assumed. We describe optimal strategies to perform data collection under various assumptions and derive corresponding time performances with respect to a simple discrete mathematical model for a sensor network. In this model, the amount of data accumulated at each sensor node (characterized by a number of unit data packets) after some given observation period is assumed finite and determined. This work is a revised and more complete version of [13] and [14], where in particular the fundamental proof of the optimality of the distribution algorithm on multiline networks was omitted. It should be noted that the presented (optimal) data collection algorithms are centralized and require cooperation between nodes which is not necessarily compatible with the requirements of sensor networks. Therefore, when requirements are more stringent, these algorithms may no longer be practical. However, they continue to provide a lower bound on the data collection time of any given collection schedule.

The following study assumes that the network achieved a stationary state and in particular that nodes have organized themselves into a network. Furthermore, we distinguish between two phases of operation in a sensory network. In the first phase or observation/measuring phase, area monitoring results in an accumulation of data at each sensor node. In the second phase or data transfer, the collected data is transmitted to some processing center located within the sensor network. In this paper, we investigate the efficiency limits with respect to time of such data transfers.

This paper is organized as follows. In Section II, we define our sensor network model. We present results in sensory networks equipped with directional antenna elements in Section III. In Section IV, we propose a generalization to omnidirectional systems. We present a comparison analysis of the two systems in

Section V. Finally, we conclude in Section VI. Pseudocode of presented algorithms and figures are grouped in the Appendices.

## II. MODEL AND PROBLEM STATEMENT

We define a sensor network as a finite collection of  $n$  identical nodes  $\{N_1, \dots, N_n\}$ . Each node  $N_i$  is associated with an integer  $\nu_i$  that represents the number of data packets stored at this node at the end of the observation period. There is one special node denoted  $N_0$ -the processing center- which we will refer to as the base station (BS). All the nodes including the BS have a common transmission range  $r$ . A node (BS included) cannot receive and transmit at the same time. The interference model defined in [8] is adopted here. That is, a transmission from node  $N_i$  to node  $N_j$  where  $i, j \geq 0$  is successful if for every other node  $N_k$ ,  $k \geq 0$  simultaneously transmitting

$$|N_i - N_j| \leq r, \quad |N_k - N_j| \geq (1 + \delta)r, \quad \delta > 0 \quad (1)$$

$(1 + \delta)r$  represents the interference range of a transmission. We assume in our model that time is slotted and a one hop transmission consumes one time slot (TS). The network is further assumed to be synchronous. A node can only transmit/receive one data packet per time slot. Multiple transmissions may occur within the network in one TS under this interference model by virtue of spatial separation. Our network may be represented as a weighted rooted graph  $\{V, E, \nu_n\}$  where  $V = \{N_0, \dots, N_n\}$ ,  $E$  denotes the set of links and  $\nu_n = (\nu_1, \dots, \nu_n)$ . In this graph model, the root represents the BS ( $N_0$ ) and an edge represents an existing wireless connection (a link) between two sensor nodes, or a sensor node and the BS. The general *data collection problem* in a given sensory network refers to the problem of routing all the data collected by the sensor nodes to the BS as efficiently as possible with respect to time and energy. However, in this paper, we shall focus on the time efficiency alone of the data collection task. In the following section, we focus our attention on directional antenna systems.

## III. DIRECTIONAL ANTENNA SYSTEMS

### A. Line Networks

In this section, we consider a line network (an example of which is given in Fig. 1). A BS is placed at one end of the network. We assume sensor nodes are regularly placed along the network. We denote by  $d$  the distance between any two nodes. Assume each node is equipped with directional antennas allowing transmissions over a distance  $r$ , where  $d < r < 2d$ . Further assume that  $\delta$  is such that  $(1 + \delta)r < 2d$ . In this scenario, there are two nodes (one on the left, one on the right) within transmission/interference range of any given node in the line (except for the end nodes). It is possible to extend this model to a more realistic scenario, where nodes are randomly placed along a line and where different values of  $r, \delta$  are considered (as long as end to end connectivity of the network is ensured). However, we find that simple case to be most insightful. More general scenarios are considered in the following section. Let  $N_i$  be the node at distance  $i$  from the BS. We denote by  $i \rightarrow i + 1$  a transmission from node  $i$  to node  $i + 1$ .

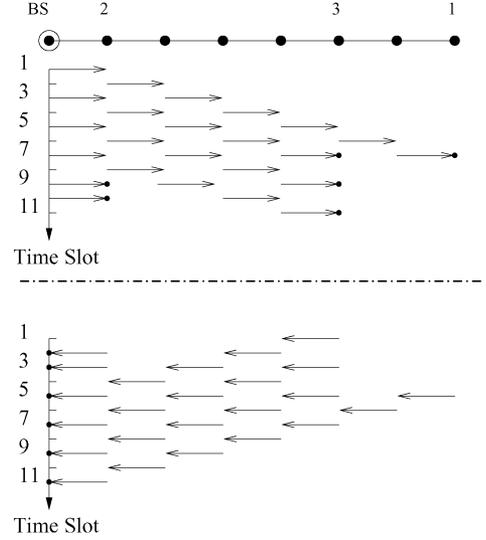


Fig. 1. 8-node line network ( $\nu_1 = 2, \nu_2 = \nu_3 = \nu_4 = \nu_6 = 0, \nu_5 = 3, \nu_7 = 1$ ) followed by optimal transmission schedules for the distribution (upper schedule) and collection (lower schedule) problems. They are symmetric of one another. The job is performed in 11 TS.

Our goal is to determine the minimal duration of the collection phase and an associated optimal communication strategy.

For the purpose of solving this problem, we look initially at the following converse problem (which we shall refer to as the *distribution problem*); instead of nodes sending packets to the BS, assume the BS is to transmit packets to nodes. The data transfer efficiency remains our concern. This problem is of separate interest in sensor networks.

We propose the following simple algorithm for solving the distribution problem. We shall prove subsequently it is optimal. The BS is to send first data packets destined for the furthest node, then data packets for the second furthest one and so on, as fast as possible, while respecting the channel reuse constraints. Nodes between the BS and its destinations are required to forward packets as soon as they arrive (that is in the TS following their arrival). We include, in Appendix II, Algorithm 1 running at the BS.

The procedure is illustrated in the example of Fig. 1, where  $V = \{0, 1, 2, 3, 4, 5, 6, 7\}$ ,  $E = \{(i, i + 1), 0 \leq i \leq 6\}$ ,  $\nu = (2, 0, 0, 0, 3, 0, 1)$ ,  $d < r < 2d$ ,  $(1 + \delta)r < 2d$ . The schedule of transmissions as determined by Algorithm 1 is drawn below the network for the distribution and collection problems, respectively. Either way it is performed in 11 TS. Next, we determine the performance of our algorithm in general. Denote by  $T_i$  the last busy TS at node  $i$  in the execution of our distribution algorithm (in the previous example, we have  $T_1 = 10, T_2 = 9, T_3 = 10, T_4 = 11, T_5 = 11, T_6 = 7, T_7 = 7$ ). Clearly, then our algorithm runs in  $\max_{1 \leq i \leq n} \{T_i\}$ .  $T_i$  is a function of the distance to the BS, the number of packets destined for node  $i$ , as well as the number of packets forwarded by node  $i$ .

*Lemma 3.1:* Assuming  $\nu_i = 0$  for  $i > n$ , node  $i$ 's last busy TS, when running Algorithm 1, is

$$T_i = \begin{cases} i - 1 + 2 \sum_{j \geq i+1} \nu_j, & \text{if } \nu_i = 0 \\ \nu_1 + 2 \sum_{j \geq 2} \nu_j, & \text{if } i = 1 \text{ and } \nu_1 \geq 1 \\ i - 2 + 2 \sum_{j \geq i} \nu_j, & \text{if } i \geq 2 \text{ and } \nu_i \geq 1 \end{cases} \quad (2)$$

*Proof:*  $\forall i \geq 1$ , node  $i$  is idle the first  $i - 1$  TS. It forwards  $\sum_{j \geq i+1} \nu_j$  data packets to further nodes and receives  $\nu_i$  data packets that are destined for itself. Forwarding a data packet consists in receiving that data packet and transmitting it right away and, therefore, a node involved in forwarding one data packet will remain busy two consecutive TS's. Receiving a data packet on the other hand consumes only one TS but in our scheme forces node  $i \geq 2$  to remain silent in the following TS. Therefore

$$\begin{aligned} \nu_1 \geq 1 &\Rightarrow T_1 = 2 \sum_{j \geq 2} \nu_j + \nu_1 \\ \nu_i \geq 1, \\ i > 1 &\Rightarrow T_i = (i - 1) + 2 \sum_{j \geq i+1} \nu_j + 2(\nu_i - 1) + 1 \\ \nu_i = 0 &\Rightarrow T_i = (i - 1) + 2 \sum_{j \geq i+1} \nu_j. \end{aligned}$$

We define, for a given sensor network,  $T_u(\boldsymbol{\nu})$  the minimum length of a time schedule over all time schedules that perform the distribution job.

*Theorem 3.2:* Assuming  $\nu_i = 0$  for  $i > n$ , the minimum collection time on directional network  $\boldsymbol{\nu}$  is

$$T_u(\boldsymbol{\nu}) = \max_{1 \leq i \leq n-1} \left( i - 1 + \nu_i + 2 \sum_{j \geq i+1}^n \nu_j \right). \quad (3)$$

*Proof:* Clearly, the maximum of  $T_i$  is obtained over the set  $\{i \geq 1 \mid \nu_i \neq 0\}$ . Thus, we have the following upper bound on  $T_u(\boldsymbol{\nu})$ :

$$T_u(\boldsymbol{\nu}) \leq \max_{\{i \geq 1 \mid \nu_i \neq 0\}} T_i.$$

A lower bound on  $T_u(\boldsymbol{\nu})$  is as follows. Assuming  $\nu_i = 0$  for  $i > n$ , we have

$$T_u(\boldsymbol{\nu}) \geq \max_{1 \leq i \leq n-1} \left( i - 1 + \nu_i + 2 \sum_{j \geq i+1}^n \nu_j \right).$$

Indeed node  $i$  has to forward  $\sum_{j \geq i+1}^n \nu_j$  data packets to further nodes. Forwarding one data packet consists in receiving and transmitting that data packet and, therefore, results in a two TS consumption (per forwarded packet). Besides, it is itself the destination of  $\nu_i$  data packets. Each received data packet costs at least one TS. Furthermore, node  $i$  cannot be active before it receives a data packet, which takes at least  $i - 1$  TS. Therefore,  $S_i \triangleq 2 \sum_{j \geq i+1} \nu_j + \nu_i + (i - 1)$  is a lower bound on any time schedule for all  $i$ . Hence,  $\max_{1 \leq i \leq n-1} S_i$  is a lower bound on  $T_u(\boldsymbol{\nu})$ .

Finally, we prove that lower and upper bounds on  $T_u(\boldsymbol{\nu})$  are equal and, therefore, the proposed schedule is optimal. Clearly,  $S_1 = T_1$  and  $\forall i \geq 2$ ,  $S_i = T_i$  if  $\nu_i \leq 1$ . On the other hand,

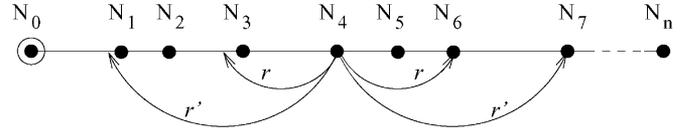


Fig. 2.  $(n + 1)$ -node line network, where  $r' = 2r$ .

if  $\nu_i > 1$ , then  $T_i > S_i$  but then either  $\nu_{i-1} = 0$  and then  $S_{i-1} = T_i$  or  $\nu_{i-1} \geq 1$  and then  $S_{i-1} > T_i$ . ■

We now return to the data collection problem. The construction of a schedule here is based on the symmetry of the operations of distribution and collection. A time schedule that is symmetric to the distribution problem's schedule with respect to a fictive horizontal axis (see example in Fig. 1) provides us with an optimal solution, the time to transmit data packets from nodes to the BS being the same as the time to carry out the converse operation (and being, therefore, minimal). In particular, a transmission  $i \rightarrow i + 1$  occurring at TS  $j$  in the distribution problem is a transmission  $i + 1 \rightarrow i$  occurring at TS  $T_u(\boldsymbol{\nu}) + 1 - j$  in the collection problem. Since the solution to one problem gives us the solution to the other, we only consider the distribution problem in the sequel. Note that an additional issue is raised in the data collection case; indeed the described algorithms do not require the network to be synchronous in the distribution case (so the algorithms may be run in a distributed way), whereas they do in the data collection case.

## B. Toward More General Scenarios

The general line case is shown in Fig. 2, although a solution to this problem is beyond the scope of this paper. It consists of  $n$  randomly located sensor nodes  $N_1, \dots, N_n$  along a line and a BS  $N_0$  at the left end of that line. It is assumed that each node's transceiver has a common transmission range  $r$  such that  $r \geq \max_{0 \leq i \leq n-1} d(N_i, N_{i+1})$  where  $d(N_i, N_{i+1})$  denotes the distance between nodes  $N_i$  and  $N_{i+1}$  (which ensures end to end connectivity of the network) and interference range  $r' = (1 + \delta)r$ . Under these assumptions any given node will have in general more than one neighbor to the right (resp. left)-those numbers varying from one node to the other.

In the rest of this section, we solve a generalized version of the line network described in the previous section in which  $r$  is fixed but  $m$  is a variable integer parameter. This allows us to study the impact of the interference range on the data collection process and provides intuition in the more general scenarios described above. We assume that the number of left and right neighbors is the same (one in this case) for all nodes for simplicity. Furthermore, it is convenient to imagine a line network with regularly spaced sensor nodes. We fix the transmission range to one hop and the interference range to  $m$  hops (that is  $r = 1$  and  $\delta = m - 1$ ). Note that in the previous section,  $m$  was taken to be 1. In practice,  $m$  is often between 2 and 3.

The distribution strategy for the BS is to transmit  $\nu_n$  data packets to node  $N_n$  first, then  $\nu_{n-1}$  packets to  $N_{n-1}$ , and so

on, as fast as possible, while respecting the channel reuse/transceiver constraints. This strategy's time performance is  $\max_i T_i$ , where

$$T_1 = \begin{cases} \sum_{1 \leq j}^{m-1} j\nu_j + m \sum_{j \geq m} \nu_j, & \text{if } \nu_1 \geq 1 \\ 2 + m \left( \sum_{j \geq m} \nu_j - 1 \right), & \text{if } \exists k \geq m \text{ such that:} \\ \quad \nu_1 = \dots = \nu_{k-1} = 0, \quad \nu_k \geq 1 \\ 2 + k(\nu_k - 1) + \sum_{k+1 \leq j}^{m-1} j\nu_j + m \sum_{j \geq m} \nu_j, & \text{if} \\ \quad \exists k 2 \leq k < m, \quad \nu_1 = \dots = \nu_{k-1} = 0, \quad \nu_k \geq 1 \end{cases}$$

$$T_2 = \begin{cases} \sum_{2 \leq j \leq m-1} j\nu_j + m \sum_{j \geq m} \nu_j, & \text{if } \nu_2 \geq 1 \\ 3 + m \left( \sum_{j \geq m} \nu_j - 1 \right), & \text{if } \exists k \geq m \text{ such that:} \\ \quad \nu_2 = \dots = \nu_{k-1} = 0, \quad \nu_k \geq 1 \\ 3 + k(\nu_k - 1) + \sum_{j \geq k+1}^{m-1} j\nu_j + m \sum_{j \geq m} \nu_j, & \text{if} \\ \quad \exists k 2 \leq k < m, \quad \nu_2 = \dots = \nu_{k-1} = 0, \quad \nu_k \geq 1 \end{cases}$$

$$T_i = \sum_{i \leq j \leq m-1} j\nu_j + m \sum_{j \geq m} \nu_j, \quad \text{if } 2 < i < m$$

$$T_{m+k} = \begin{cases} k + m \sum_{j \geq m+k} \nu_j, & \text{if } \nu_{m+k} \geq 1, \\ k + 1 + m \sum_{j \geq m+k} \nu_j, & \text{if } \nu_{m+k} = 0, \quad \text{if } k \geq 0. \end{cases}$$

The proof follows a similar argument as the one used to prove Lemma 3.1 and is omitted. The following theorem gives a closed form expression for the minimum data collection delay. This generalizes Theorem 3.2.

**Theorem 3.3:** The minimum data collection time  $T_u^m(\boldsymbol{\nu})$  on directional line network  $\boldsymbol{\nu}$  when the interference range is  $m$  times the transmission range is

$$T_u^m(\boldsymbol{\nu}) = \begin{cases} \max_i \left( i - 1 + \sum_{j \geq i}^{i+m-2} (j - i + 1) \nu_j \right. \\ \quad \left. + m \sum_{j \geq i+m-1} \nu_j, \quad \forall m \geq 2 \right. \\ \left. \max_i \left( i - 1 + \nu_i + 2 \sum_{j \geq i+1} \nu_j \right), \quad \text{if } m = 1. \right. \end{cases} \quad (4)$$

*Proof:* We have  $T_u^m(\boldsymbol{\nu}) \leq \max_i T_i$ . A lower bound on the minimum time performance can be derived as well

$$\forall i \geq 1, \\ T_u^m(\boldsymbol{\nu}) \geq i - 1 + \sum_{j \geq i} \nu_j + \sum_{j \geq i+1} \nu_j + \dots + \sum_{j \geq i+m-1} \nu_j. \quad (5)$$

Indeed transmissions  $i - 1 \rightarrow i, i \rightarrow i + 1, \dots, i + m - 2 \rightarrow i + m - 1$  may not occur concurrently due to channel reuse constraints. Inequality (5) may be rewritten

$$\forall m \geq 2, \quad T_u^m(\boldsymbol{\nu}) \geq \max_i (S_i) \quad (6)$$

where

$$S_i \triangleq \left( i - 1 + \sum_{i \leq j \leq i+m-2} (j - i + 1) \nu_j + m \sum_{j \geq i+m-1} \nu_j \right). \quad (7)$$

The case  $m = 1$  may be derived from the above formula by choosing  $m = 2$ . Assume there exists  $j_0, 1 \leq j_0 \leq n$  such that  $\forall i \neq j_0, T_{j_0} \geq T_i, T_{j_0+1} < T_{j_0}$ .

- If  $j_0 = 1$ , then  $\nu_1 \geq 1 \Rightarrow S_1 = T_1$ .

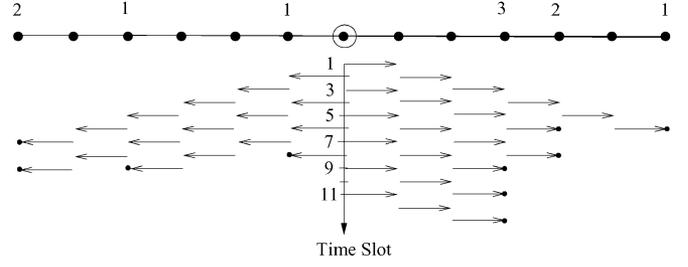


Fig. 3. Optimal distribution schedule on a two-line sensor network. The job is performed in 13 TS.

- If  $j_0 = 2$ , then  $\nu_2 \geq 1, \nu_1 = 0 \Rightarrow T_2 - S_2 = \sum_{2 \leq j \leq m} \nu_j - 1 \geq 0, \nu_1 = 0 \Rightarrow T_1 = T_2 \Rightarrow S_1 \geq T_2$ . Indeed  $S_1 \geq T_1$  since

$$\nu_1 \geq 1 \Rightarrow S_1 = T_1$$

and

$$\nu_1 = 0 \Rightarrow S_1 - T_1 = \begin{cases} \sum_{1 \leq j \leq m-1} j\nu_j + m - 2 \geq 0 \quad (m \geq 2) \text{ or} \\ \sum_{1 \leq j \leq k-1} j\nu_j + k - 2 \geq 0 \quad (k \geq 2) \end{cases}$$

- If  $2 < j_0 < m \Rightarrow \nu_i \geq 1, \nu_1 = \dots = \nu_{i-1} = 0 \Rightarrow T_{j_0} - S_1 = -\sum_{1 \leq j \leq i-1} j\nu_j = 0$ .
- If  $j_0 = m+k, k \geq 0 \Rightarrow \nu_{m+k} \geq 1, \nu_k = \dots = \nu_{k+m-1} = 0 \Rightarrow T_{m+k} - S_{k+1} = -\sum_{j \geq k+1}^{k+m-1} (j - k) \nu_j = 0$ .

Therefore,  $\max_i T_i = \max_i S_i$  and Theorem 3.3 follows. ■

### C. Two-Line Networks

Consider now a line network and place the BS anywhere on that line. This may be seen as a two-line network  $(\boldsymbol{\mu}, \boldsymbol{\nu})$ . We denote by  $T_u(\boldsymbol{\mu}, \boldsymbol{\nu},)$  the optimal performance achievable on a two-line network. The scheduling procedure, a particular case of the multiline algorithm described in the next section, is illustrated in the example of Fig. 3.

**Theorem 3.4:** The minimum collection time on a directional two-line network  $(\boldsymbol{\mu}, \boldsymbol{\nu})$  is

$$T_u(\boldsymbol{\mu}, \boldsymbol{\nu}) = \begin{cases} \max \left( T_u(\boldsymbol{\nu}) + 1, \sum_{i \geq 1} \mu_i + \nu_i \right), \\ \quad \text{if } T_u(\boldsymbol{\mu}) = T_u(\boldsymbol{\nu}) \\ \max \left( T_u(\boldsymbol{\nu}), \sum_{i \geq 1} \mu_i + \nu_i \right), \\ \quad \text{if } T_u(\boldsymbol{\nu}) > T_u(\boldsymbol{\mu}). \end{cases} \quad (8)$$

*Proof:* See [12]. ■

### D. Multiline Networks

The algorithm running at the BS (listed as Algorithm 2 in Appendix II) determines at each TS toward which line to transmit, if transmission is possible at all. The direction of transmission is greedily decided, based on estimates (one per line) of the completion time of the data transfer. Initial estimate for a given line is determined by (3). The legal direction associated with the biggest estimate is chosen (a legal transmission is one that respects the channel reuse constraints. For example, it is not legal for our algorithm to transmit in two successive TS's toward a given node located at distance greater than 1 from the BS), ties being broken randomly. When no legal direction exists

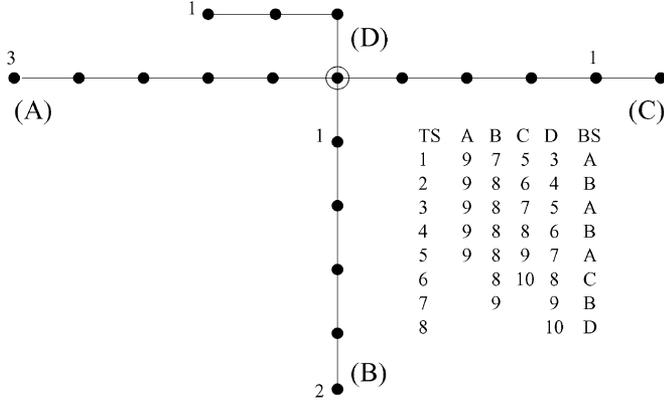


Fig. 4. Optimal distribution schedule for BS on a four-line sensor network. The completion time is 10 TS.

the BS remains idle. After a decision has been made (transmit toward a particular direction or stay idle) the estimates at each line are updated according to the following rule: If a legal direction was not chosen, its new estimate becomes its old estimate plus one. Illegal direction estimates remain unchanged. The idea is to minimize at each TS the overall estimate of the transmission time.

We illustrate the procedure on an example in Fig. 4. In the accompanying table, we list data transfer completion time estimates at each TS and the corresponding decision made by the BS. As previously stated, the initial completion time estimates are computed using (3). The table reads as follows. TS 1: All four transmission directions are legal. The BS chooses to transmit toward line *A*. At TS 2, transmitting toward *A* is not a legal move, the legal transmission direction associated with the biggest estimate is *B*, etc. Along a given line, the packets destined for furthest nodes are sent first by the BS. As for the other nodes, they merely forward the data packets of which they are not recipients (a packet is transmitted in the following TS that it was received). In this example, the algorithm performs in 10 TS.

*Theorem 3.5:* Algorithm 2 is optimal.

*Proof:* We note that equivalently this algorithm picks at each TS the legal direction  $B_i$  that maximizes  $T(\mathbf{v}^i)$  (that quantity being updated at each TS to take into account the packets delivered).

We first introduce a few notations and definitions. Let  $N$  denote the considered network for which one wishes to derive an optimal schedule. Let  $N_e$  denote the “equivalent” network to  $N$  (see following definition). Let  $P$  denote the considered problem of scheduling data transfers to the nodes. Let  $P'$  denote the same problem under a relaxed set of conditions, namely that simultaneous transmission and reception (of different data packets) are allowed in a single TS at any given node. This problem is independently studied in Appendix I. Let  $S(P, N)$  denote a schedule for problem  $P$  and network  $N$ . Let  $S_{|BS}(P, N)$  denote the schedule of the BS derived from  $S(P, N)$ . Let  $S^{\text{opt}}(P, N)$  denote an optimal schedule for  $(P, N)$ .

In the “equivalent” network  $N_e$  of network  $N$ , the data packets along a particular line are redistributed along the corresponding line in  $N_e$  in the manner illustrated in Fig. 5. Although packets in  $N$  and  $N_e$  are distributed differently over

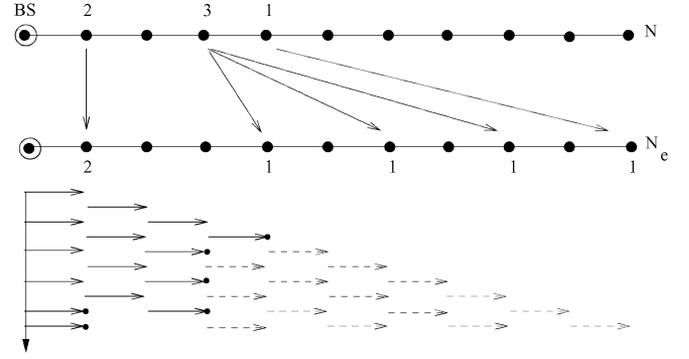


Fig. 5. “Equivalent” network transformation. Packets along a given line in  $N$  are redistributed (away from the BS) along the corresponding line in  $N_e$  so that the distribution schedules for both lines have the same length. The distribution schedules for that particular line in network  $N$  (solid arrows) and its corresponding line in network  $N_e$  (solid arrows + dashed arrows), respectively, are drawn. The latter schedule is an extension of the former.

the network, we shall see that the data collection is the same for both networks. It is in that sense that they are “equivalent.” Formally, the transformation is as follows. To each line of  $N$ , say  $B_k$ , if  $\nu_i^k$  denotes the number of data packets at distance  $i$  from the BS along  $B_k$ ,  $T(\mathbf{v}^k)$  denotes the length of an optimal schedule for that particular line and  $T_i^k$  is the last busy TS at node  $i$  in the execution of Algorithm 1 for that line, associate a line in  $N_e$ , say  $B'_k$ , such that if  $\nu_i^k$  denotes the number of data packets at distance  $i$  from the BS along  $B'_k$

$$\begin{aligned}
 i = 1 & \quad \nu_1^k = \nu_1^k \\
 i \geq 2 & \quad \nu_{T(\mathbf{v}^k) - T_i^k + 2j + i}^k = 1 \\
 & \quad \text{for } 0 \leq j \leq l - 1, \text{ if } \nu_i^k = l \geq 1 \\
 \nu_i^k & = 0, \quad \text{otherwise.}
 \end{aligned} \tag{9}$$

By construction,  $N_e$  has the following characteristics.

- Same total number of data packets as  $N$ , same number of data packets per line, and same number of lines.
- Each line carries the same workload as its corresponding line in  $N$  (i.e.,  $\forall k, T(\mathbf{v}^k) = T(\mathbf{v}^k)$ ).
- Node  $i > 1$  carries 0 or 1 data packet.
- Two nodes with data packets are separated by at least one node with no data packet.

*Example:* Consider the following two-line network  $N$ :  $B_1 : \mathbf{v}^1 = (0, 4)$ ,  $B_2 : \mathbf{v}^2 = (2, 0)$ . Its equivalent network  $N_e$  is a two-line network such that:  $B'_1 : \mathbf{v}^1 = (0, 1, 0, 1, 0, 1, 0, 1)$ ,  $B'_2 : \mathbf{v}^2 = (2, 0) = \mathbf{v}^2$ .

*Lemma 3.6:* There exists an optimal schedule for problem  $P'$  and network  $N_e$ .

*Proof:* A construction of such a schedule is given in Appendix I. ■

*Lemma 3.7:* Let  $S^{\text{opt}}(P', N_e)$  denote the optimal schedule constructed in Appendix I for problem  $P'$  and network  $N_e$ . It is possible to construct a schedule  $S(P, N_e)$  for problem  $P$  and network  $N_e$  from  $S^{\text{opt}}(P', N_e)$  by judiciously reordering the BS transmissions such that the two schedules have the same length.

*Proof:* With the convention that furthest nodes should be served first along a given line, a schedule  $S(P/P', N_e)$

is entirely defined by its restriction to the BS schedule  $S_{|BS}(P/P', N_e)$ . The BS schedule being a sequence of directions  $B_i$  corresponding to the lines toward which transmit at each TS, as well as possible silences (corresponding to BS being idle).  $S_{|BS}(P', N_e) = (B'_1, B'_1, B'_1, B'_1, B'_2, B'_2, -, -)$  is an instance of an optimal schedule for problem  $P'$  and the network described in previous example where “-” denotes a silence.

We construct  $S(P, N_e)$  from  $S^{\text{opt}}(P', N_e)$  by iteratively applying the following operation on  $S$ :  $\text{insert}(i, j)(S)$  for  $j > i \geq 1$ , which returns a schedule  $S'$ , where element  $j$  in schedule  $S$  was inserted between element  $i$  and  $i + 1$  in  $S$ . In the previous example,  $\text{insert}(1, 5)(S_{|BS}(P', N_e)) = (B'_1, B'_2, B'_1, B'_1, B'_1, B'_2, -, -)$ .

This operation does not change the length of  $S$ , that is  $\text{Length}(S') = \text{Length}(S)$  as long as it is not applied more than once for any  $i$ . This is a direct consequence of the fourth characteristic of an equivalent network. Next, we describe the construction.

If  $S$  is a valid schedule (i.e., satisfying constraint  $P$ ) we are done. Otherwise, assume first conflict occurs in position  $i_0$  of schedule  $S$  (that is constraint  $P$  does not allow for transmission toward element  $i_0$  followed by transmission toward element  $i_0 + 1$ ). In the instance above, there are conflicts in  $i_0 = 1, 2, 3, 5$ . Further, assume the first direction distinct from the one in position  $i_0$  and that follows it is element  $i_1$  of  $S$ . If there is no such direction, then denote  $i_1$  the position of the first silence following (it always exists by definition of  $N_e$ ). Then, apply  $\text{insert}(i_0, i_1)(S)$ . Clearly, the procedure produces a new schedule  $S$  of same length. Thus, the portion of the schedule  $S$  comprised between element 1 and  $i_0 + 1$  satisfies  $P$ . Repeat until the schedule  $S$  satisfies constraint  $P$ . Since the number of initial conflicts is finite, this procedure ends in a finite number of steps. In the previous example, these operations are in order:  $\text{insert}(1, 5)$ ,  $\text{insert}(3, 6)$ , and  $\text{insert}(5, 7)$ . They lead to the schedule:  $S_{|BS}(P, N_e) = (B'_1, B'_2, B'_1, B'_2, B'_1, -, B'_1, -)$ . ■

By Lemma 3.7, the lengths of  $S(P, N_e)$  and  $S^{\text{opt}}(P', N_e)$  are the same. Thus,  $S(P, N_e)$  is optimal. Denote it  $S^{\text{opt}}(P, N_e)$ . One may construct a schedule  $S(P, N)$  from  $S^{\text{opt}}(P, N_e)$  such that the lengths of the two schedules are the same and  $S_{|BS}(P, N) = S_{|BS}(P, N_e)$ . If  $S(P, N)$  is not optimal, then there exists a schedule  $S'(P, N)$  such that the length of  $S'(P, N)$  is less than the length of  $S(P, N)$ , but from  $S'(P, N)$  one may construct a schedule  $S'(P, N_e)$  such that the two schedules have the same length and  $S'_{|BS}(P, N_e) = S'_{|BS}(P, N)$  so the length of  $S'(P, N_e)$  is less than the length of  $S^{\text{opt}}(P, N_e)$ , a contradiction. Thus,  $S(P, N)$  is optimal, which concludes the proof of Theorem 2. ■

### E. Tree Networks, Case Where Degree of Base Station Is One

Throughout this paragraph, we assume that the degree of the root of the considered graphs is one. We define the *equivalent linear network*  $(G_l, E_l, \nu_l)$  of a network  $(G, E, \nu)$ : If  $G = \{N_0, N_1, \dots, N_n\}$  and  $\nu = (\nu_1, \dots, \nu_n)$ , then  $G_l = \{0, 1, \dots, m\}$ ,  $E_l = \{(i-1, i), 1 \leq i \leq m\}$ , and  $\nu_l = (\nu_{l1}, \dots, \nu_{lm})$ , where  $m = \max_i(d(N_0, N_i))$  and  $\nu_{lj} = \sum_{i|d(N_0, N_i)=j} \nu_i$ . We illustrate a tree network in Fig. 6

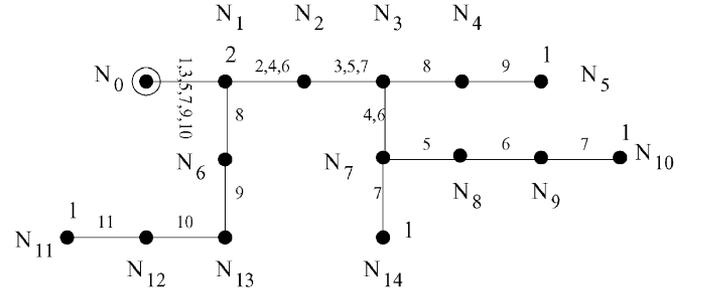


Fig. 6. A 15-node tree network with degree of BS = 1, the equivalent linear network is drawn in Fig. 1. Transmission TS are written next to the edges.

( $n = 14, m = 7$ ); its equivalent linear network is shown in Fig. 1.

The equivalent linear network's schedule may serve as a schedule for the initial tree network. Next, we explain how transmission TS's for  $(G_l, E_l, \nu_l)$  (determined by running algorithm 1) may be mapped onto  $(G, E, \nu)$ . Consider an element in  $E$ , say  $(N_{i_0}, N_{j_0})$ , such that  $d(N_0, N_{i_0}) = \alpha$  (hops). Based on the number of data packets  $N_{j_0}$  has to forward, say  $f_{j_0}$ , we shall allocate transmission time slots to edge  $(N_{i_0}, N_{j_0})$ . Define  $E_\alpha = \{(N_i, N_j) \in E | d(N_0, N_i) = \alpha\}$ . Each packet  $P$  follows a path  $\text{path}(P)$  from the BS to its destination node, where  $\text{path}(P)$  denotes the finite sequence of edges  $(e_1, \dots, e_k)$  traversed in that order by  $P$ . For convenience, we shall write  $\text{path}(P)$  as the sequence of vertices  $(\text{vertices}(e_1), \dots, \text{vertices}(e_k))$ . We define  $\mathfrak{P}_\alpha = \{P | \exists e \in E_\alpha \cap \text{path}(P)\}$ . We define  $\mathfrak{T}_\alpha = \{\text{TS used by } (\alpha, \alpha + 1) \in E_l\}$ . We have  $|\mathfrak{P}_\alpha| = \sum_{(N_i, N_j) \in E_\alpha} (\nu_j + f_j) = \sum_{k > \alpha} \nu_{lk} = |\mathfrak{T}_\alpha|$ . Thus, one may define a one-to-one correspondence  $g$  between  $\mathfrak{P}_\alpha$  and  $\mathfrak{T}_\alpha$  that associates the packet  $P$  with the longest path in  $\mathfrak{P}_\alpha$ , with the TS with the smallest index in  $\mathfrak{T}_\alpha$ ; the packet  $P$  with second longest path, with the TS with second smallest index and so on. We finally define  $\mathfrak{P}_\alpha^{(N_{i_0}, N_{j_0})} = \{P | (N_{i_0}, N_{j_0}) \in \text{path}(P)\} \subseteq \mathfrak{P}_\alpha$ .  $(N_{i_0}, N_{j_0})$  is associated with TS's  $g(\mathfrak{P}_\alpha^{(N_{i_0}, N_{j_0})})$ . In the example of Fig. 6, we have  $\{P\} = \{P_1, P_2, \dots, P_6\}$ , where the first packet is characterized by  $\text{path}(P_1) = (N_0, N_1, N_2, N_3, N_7, N_8, N_9, N_{10})$ , the second one by  $\text{path}(P_2) = (N_0, N_1, N_2, N_3, N_4, N_5)$ , the third one by  $\text{path}(P_3) = (N_0, N_1, N_6, N_{13}, N_{12}, N_{11})$ , the fourth one by  $\text{path}(P_4) = (N_0, N_1, N_2, N_3, N_7, N_{14})$ , and finally the fifth and sixth ones by  $\text{path}(P_5) = \text{path}(P_6) = (N_0, N_1)$ . We also have  $E_1 = \{(N_1, N_2), (N_1, N_6)\}$ ,  $\mathfrak{P}_1 = \{P_1, P_2, P_3, P_4\}$ ,  $\mathfrak{T}_1 = \{2, 4, 6, 8\}$ , and  $\mathfrak{P}_1^{(N_1, N_2)} = \{P_1, P_2, P_4\}$ . Thus, edge  $(N_1, N_2)$  is associated with TS's  $g(\mathfrak{P}_1^{(N_1, N_2)}) = \{2, 4, 6\}$ . Thus, Algorithm 1 run on the equivalent linear network provides a BS transmission schedule. Intermediate nodes simply forwards data packets to further nodes as they arrive (in the TS following their arrival). This requires a routing table at junction nodes. In a centralized version of this algorithm, nodes may be informed of their transmission slots. Fig. 4 shows such a mapping for the considered example.

Although an equivalent linear network has a reduced set of possible concurrent transmissions, this procedure produces an

optimal transmission schedule. This follows from the following lemma.

*Lemma 3.8:* Given any connected graph  $G$  such that degree of BS is one, if  $t_2(G)$  denotes the time performance of a given data distribution algorithm, and  $\nu_j$  denotes the number of data packets at distance  $j$  from the BS, then

$$t_2(G) \geq \max_i \left( i - 1 + \nu_i + 2 \sum_{j>i} \nu_j \right). \quad (10)$$

*Proof:*  $\sum_{j \geq 1} \nu_j$  data packets must be delivered to nodes at distance greater than 1. Therefore, link (0, 1) is activated  $\sum_{j \geq 1} \nu_j$  times and links (1, 2) (all edges from a node at distance 1 from the BS to a node at distance 2) are activated  $\sum_{j \geq 2} \nu_j$  times but link (0, 1) cannot be activated at the same time as a link (1, 2), thus, we have  $t_2(G) \geq \sum_{j \geq 1} \nu_j + \sum_{j \geq 2} \nu_j$ .

$\sum_{j \geq i} \nu_j$  data packets must be delivered to nodes at distance greater than  $i > 1$ . Therefore, edge (0, 1) is activated at least  $\sum_{j \geq i} \nu_j$  times and edges (1, 2)  $\sum_{j > i} \nu_j$  times but link (0, 1) cannot be activated at the same time as a link (1, 2), moreover after  $\sum_{j \geq i} \nu_j + \sum_{j > i} \nu_j$  TS the last data packet sent by the BS is at distance 0 or 1 from the BS if  $\nu_i > 0$  and at distance 0, 1, or 2 from the BS if  $\nu_i = 0$ . Indeed it takes a minimum of  $2 \sum_{j \geq i} \nu_j$  TS to get all the data packets out of the positions 0, 1, and 2. Thus, after  $\sum_{j \geq i} \nu_j + \sum_{j > i} \nu_j$  TS whether  $\nu_i > 0$  or  $\nu_i = 0$  one data packet is at least  $i - 1$  hop away from its destination, therefore,  $t_2(G) \geq \sum_{j \geq i} \nu_j + \sum_{j > i} \nu_j + i - 1$ . Hence, the stated result. ■

#### F. Tree Sensor Networks, General Case

The results in the previous sections suggest the following algorithm for dealing with general tree networks.

- 1) Linearize the subtrees attached to the BS according to the procedure described in Section III-E.
- 2) Apply multiline algorithm described in Section III-D to the resulting multiline system.

This procedure produces an optimal schedule. This results from Theorem 2 and results from Section III-E.

*Theorem 3.9:* If  $T$  is a tree network and  $\nu_j^k$  denotes the number of data packets at distance  $j$  from the BS along branch  $k$ , then the minimum data collection time over  $T$  is

$$T_u(T) = \max_{1 \leq i \leq n} \left( i - 1 + \sum_{j \geq i} \nu_j' \right) \quad (11)$$

where  $\nu_j' = \sum_k \nu_j^k$  and  $\nu_j^k$  is obtained from  $\nu_j^k$  by (9).

*Proof:* This follows directly from Lemma 3.7. ■

#### G. Networks With Cycles

We propose a data distribution/collection strategy on general graphs. However, this strategy is not optimal in general. In this section, we prove that our algorithm performs within a factor of 2 of an optimal strategy. The proposed strategy consists of two subprocedures.

- 1) Extract a shortest path spanning tree  $\mathcal{T}_{SP}$ .

- 2) Apply previously described distribution strategy on trees to  $\mathcal{T}_{SP}$ .

Note: One can show that shortest path spanning trees always exist by using Dijkstra algorithm. The following theorem provides a motivation for choosing a shortest path spanning tree. The proof follows from Theorem 3.9.

*Theorem 3.10:* For any (connected) graph  $G$ , for any spanning tree  $\mathcal{T}$  of  $G$  and for any shortest path spanning tree  $\mathcal{T}_{SP}$  of  $G$ , the minimum data collection time over network  $\mathcal{T}$ ,  $T_u(\mathcal{T})$  satisfies

$$T_u(\mathcal{T}_{SP}) \leq T_u(\mathcal{T}). \quad (12)$$

*Theorem 3.11:* For any (connected) graph  $G$ , and any shortest path spanning tree  $\mathcal{T}_{SP}$ , we have

$$\frac{T_u(\mathcal{T}_{SP})}{2} \leq T_u(G) \leq T_u(\mathcal{T}_{SP}). \quad (13)$$

*Proof:* The second inequality is clear. For a proof of the first inequality we define:  $t_1(G)$  the minimum distribution time when transmission and reception are simultaneously allowed in a TS at any given node. Clearly,  $t_1(G) \leq T_u(G)$ . By corollary 1.4 (Appendix I), we also have  $t_1(G) = t_1(\mathcal{T}_{SP})$ . Besides, for any connected graph  $A$  the following inequality holds  $T_u(A) \leq 2t_1(A)$ . Choose  $A = \mathcal{T}_{SP}$ , the inequality follows. ■

These bounds are tight. The upper bound is achieved when  $G = \mathcal{T}_{SP}$ . As for the lower bound consider the following network  $G$ , where  $n$  data packets are stored at distance  $k$  hops from the BS in node  $x$ . Further, assume there are two distinct paths of length  $k$  from the BS to  $x$ .

$\mathcal{T}_{SP}$  is the line network  $\nu = \overbrace{(0, 0, \dots, 0, n)}^{k-1}$ . We have  $T_u(G) = n + k - 1$  (for  $k \geq 1$ ) and  $T_u(\mathcal{T}_{SP}) = 2n + k - 2$  (for  $k \geq 2$ ), thus,  $T_u(G)$  converges toward  $T_u(\mathcal{T}_{SP})/2$  when  $n$  goes to infinity (for  $k \geq 2$ ).

Bounds on  $T_u(G)$  can also be written in the following more explicit way.

*Theorem 3.12:* The minimum data collection time over a graph  $G$  satisfies

$$\max_i \left( i - 1 + \sum_{j \geq i} \nu_j \right) \leq T_u(G) \leq \max_i \left( i - 1 + \nu_i + 2 \sum_{j \geq i+1} \nu_j \right). \quad (14)$$

*Proof:* We have from Corollary 1.4 (Appendix I)  $t_1(G) = \max_i (i - 1 + \sum_{j \geq i} \nu_j)$ . ■

Both bounds on  $T_u(G)$  are achievable. The lower bound for instance is achieved in the previously considered example, where  $\max_i (i - 1 + \sum_{j \geq i} \nu_j) = n + k - 1$ .

## IV. OMNIDIRECTIONAL ANTENNA SYSTEMS

### A. Line Networks

Our results readily extend to omnidirectional antenna systems. We have, for a line network the following result.

*Theorem 4.1:* The minimum data collection time over an omnidirectional line network  $\nu$ , assuming the interference range is  $m$  times the transmission range, is

$$\forall m \geq 1$$

$$T_o^m(\nu) = \max_i \left( i - 1 + \sum_{i \leq j \leq i+m} (j - i + 1) \nu_j + (m + 2) \sum_{j \geq i+m+1} \nu_j \right). \quad (15)$$

*Proof:* The proof follows a similar argument as the one used to prove Theorem 4.1. ■

## V. OMNIDIRECTIONAL/DIRECTIONAL ANTENNA SYSTEMS COMPARISON

The following result compares the performance of omnidirectional and directional antenna systems over a single line network.

*Theorem 5.1:* For any line network  $\nu$  the ratio of minimum data collection times over a line network, assuming interference range is  $m$  times the transmission range, satisfies

$$1 \leq \frac{T_o^m(\nu)}{T_u^m(\nu)} < \begin{cases} 1.5, & m = 1 \\ 1 + \frac{2}{m}, & m \geq 2 \end{cases}.$$

*Proof:* The case  $m = 1$  is proven in [13]. In the case  $m \geq 2$ , assume there exists  $j_0, 1 \leq j_0 \leq n$  such that for all  $i, i \neq j_0$   $T_{j_0} \geq T_i$  and  $T_{j_0+1} < T_{j_0}$ . From Theorems 3.3 and 4.1 it follows.

Case 1)

$$j_0 = m + 2 + k \Rightarrow T_{j_0}^o = S_{k+1}$$

$$T_{j_0}^u = S_{k+3} \Rightarrow \frac{T_o^m(\nu)}{T_u^m(\nu)} = \frac{k + \sum_{j=k+1}^{k+1+m} (j - k) \nu_j + (m + 2) \sum_{j \geq k+m+2} \nu_j}{k + 2 + \sum_{j=k+3}^{k-1+m} (j - k - 2) \nu_j + m \sum_{j \geq k+m+2} \nu_j}$$

$$j_0 = m + 2 + k \Rightarrow \nu_{k+1} = \dots = \nu_{k+1+m} = 0 \Rightarrow \frac{T_o^m(\nu)}{T_u^m(\nu)} = \frac{k + (m + 2) \sum_{j \geq k+m+2} \nu_j}{k + 2 + m \sum_{j \geq k+m+2} \nu_j} < \frac{m + 2}{m}.$$

Case 2)

$$j_0 = m + 1 \Rightarrow T_{j_0}^o = S_1$$

$$T_{j_0}^u = S_2 \Rightarrow \frac{T_o^m(\nu)}{T_u^m(\nu)} = \frac{\sum_{j=1}^{m+1} j \nu_j + (m + 2) \sum_{j \geq m+2} \nu_j}{1 + \sum_{j=2}^m (j - 1) \nu_j + m \sum_{j \geq m+1} \nu_j}$$

$$j_0 = m + 1 \Rightarrow \nu_1 = \dots = \nu_m = 0 \Rightarrow \frac{T_o^m(\nu)}{T_u^m(\nu)} = \frac{(m + 1) \nu_{m+1} + (m + 2) \sum_{j \geq m+2} \nu_j}{1 + m \nu_{m+1} + m \sum_{j \geq m+2} \nu_j} < \frac{m + 2}{m}.$$

Case 3)

$$1 \leq j_0 < m \Rightarrow T_{j_0}^o = S_1$$

$$T_{j_0}^u = S_1 \Rightarrow \frac{T_o^m(\nu)}{T_u^m(\nu)} = \frac{\sum_{j=1}^{m+1} j \nu_j + (m + 2) \sum_{j \geq m+2} \nu_j}{\sum_{j=1}^{m-1} j \nu_j + m \sum_{j \geq m} \nu_j} < \frac{m + 2}{m}.$$

Note: Bounds in Theorem 5.1 are tight. This is clear in the case of the lower bound. As for the upper bound, consider  $\nu = \mathbf{1}_n$  (case  $m = 2$ ), then, we have

$$\frac{T_o^m(\nu)}{T_u^m(\nu)} = \frac{\left( \sum_1^{m+1} j + (m + 2) \sum_{m+2}^n 1 \right)}{\sum_1^{m-1} j + m \sum_m^n 1} = \frac{(m + 2)(m + 1)/2 + (m + 2)(n - m - 1)}{m(m - 1)/2 + m(n - m + 1)} \xrightarrow{n} \frac{m + 2}{m}.$$

## VI. CONCLUSION

This paper is concerned with the delay in collecting at the BS, data from sensory networks. The minimum data collection time on tree networks was derived and corresponding optimal scheduling strategies were described. We focused our analysis on systems equipped with directional antennas and showed that more realistic hypotheses could be incorporated in our model (at the expense of the simplicity of the analysis). The study of omnidirectional antenna systems follows under the same lines and performances of the two systems were compared on a simple scenario. Finally, graphs with cycles were considered and the performance of our algorithms on such graphs were compared with the optimal achievable performance. This lead to bounds on the minimum time performance of optimal data collection strategies for general graphs.

## APPENDIX I

### PRELIMINARY RESULTS FOR SECTION III-D

In the following section, we assume that a network equipped with directional nodes may receive *and* transmit a data packet during any given TS (whereas so far, we had assumed that it was only possible to receive *or* transmit a data packet in a given TS). Although such networks may seem artificial and not practical for the time being, the results that follow allow us to gain some insight into more complex systems.

The purpose of this section is the construction of an optimal strategy for collecting data, as well as deriving a closed form expression for time performance. We obtain both for *any* general connected graphs. To that end, as in Section III, we are first going through a series of successive building steps.

### A. Lower Bound on the Time Performance of Data Distribution Algorithms

*Lemma 1.1:* Given any connected graph  $G$ , if  $t_1(G)$  denotes the time performance of a given data distribution algorithm, and

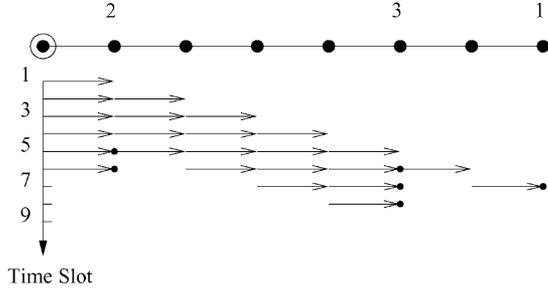


Fig. 7. Optimal distribution schedule for BS on a line network equipped with directional antennas and ability to receive and transmit in the same TS. The completion time is 8 TS.

$\nu_j$  denotes the number of data packets at distance  $j$  from the BS, then

$$t_1(G) \geq \max_i \left( i - 1 + \sum_{j \geq i} \nu_j \right). \quad (16)$$

*Proof:*  $\sum_{j \geq 1} \nu_j$  data packets must be delivered to nodes at distance greater than 1. Since the BS can only transmit one data packet at a time, we have  $t_1(G) \geq \sum_{j \geq 1} \nu_j$ .

$\sum_{j \geq i} \nu_j$  data packets must be delivered to nodes at distance greater than  $i > 1$ . After  $\sum_{j \geq i} \nu_j$  TS the last data packet sent by the BS is at distance one from the BS and, therefore, at least  $i - 1$  extra TS are required for it to reach its destination, thus,  $t_1(G) \geq \sum_{j \geq i} \nu_j + i - 1$ . Hence, the stated result. ■

### B. Achievability of Lower Bound

1) *Line Network:* The purpose of this section is to prove that the lower bound derived in the previous section is achievable on a line network. We shall show in the next section achievability on general connected graphs based on this result.

*The Algorithm:* The BS is to send first data packets destined for the furthest node, then data packets for the second furthest one and so on, as fast as possible, while respecting the channel reuse constraints. Nodes between the BS and its destinations are required to forward packets as soon as they arrive (that is in the TS following their arrival). This algorithm is illustrated by an example on Fig. 7.

*Proof of Optimality and Time Performance:* Denote  $T_i$  the last busy TS at node  $i$  in the execution of our algorithm. Clearly, then our algorithm runs in  $\max_{1 \leq i \leq n} \{T_i\}$ .  $T_i$  is a function of the distance to the BS, the number of data packets destined for node  $i$  and the number of data packets forwarded by node  $i$ .

*Lemma 1.2:*

$$T_i = \begin{cases} i + \sum_{j > i} \nu_j, & \text{if } \nu_i \leq 1 \\ i - 1 + \sum_{j \geq i} \nu_j, & \text{if } \nu_i \geq 1 \end{cases}. \quad (17)$$

*Proof:*

$$\nu_i \leq 1 \Rightarrow T_i = (f_i + 1) + (i - 1)$$

$$\nu_i > 1 \Rightarrow T_i = (f_i + 1) + \nu_i - 1 + (i - 1)$$

$$f_i = \text{number of packets forwarded by } i = \sum_{j > i} \nu_j.$$

*Lemma 1.3:* Define:  $S_i = \sum_{j \geq i} \nu_j + i - 1$ , then  $\max_i S_i = \max_i T_i$

*Proof:* Indeed  $S_i$  is a lower bound for all  $i$ . So  $\max_i S_i \leq \max_i T_i$ , but  $S_i = T_i$  if  $\nu_i \geq 1$ . Since clearly  $\max_i T_i$  occurs in  $i$  such that  $\nu_i \geq 1$ , we have  $\max_i S_i = \max_i T_i$ , i.e., the algorithm is optimal. ■

2) *General Connected Graphs:* By using the shortest routes (from the BS) to the sensor nodes, the algorithm previously described on line networks may be used on general (connected) graphs. The performance time of that algorithm is then  $\max_i T_i$ , where  $T_i$  is defined in Lemma 1.3 and  $\nu_j$  is the number of data packets at distance  $j$  from the BS. The next corollary follows from Lemma 1.3.

*Corollary 1.4:* The minimum data collection time  $t_1(G)$  on any connected graph  $G$  is

$$t_1(G) = \max_{1 \leq i \leq n} \left( i - 1 + \sum_{j \geq i} \nu_j \right). \quad (18)$$

The following corollary follows from Corollary 1.4.

*Corollary 1.5:*

$$\forall \mathcal{T} \text{ a spanning tree of } G, \quad t_1(\mathcal{T}_{SP}) \leq t_1(\mathcal{T}). \quad (19)$$

## APPENDIX II ALGORITHMS

Given a line network (represented by the vector  $\text{Network} = \nu$ ), Algorithm 1 dictates the BS actions at each time step: remain idle (action = 0) or transmit (action = 1). The result is stored in the vector  $\text{action}$ . When an action is chosen the right packet is to be handed over to the BS for transmission. One might assume that there is a stack of data packets correctly ordered with respect to the distance to the BS and that the stack is being updated after each BS action so that a packet is popped off the stack as it is transmitted.

**Algorithm 1:** Determines BS actions in line networks

**input:** Network

**output:** action

```

1: step ← 1, legal ← 1, packets_left ← 1 ∑i Network(i)
2: while packets_left ≠ 0 do
3:   if legal then
4:     action(step) ← 1
5:     packets_left ← packets_left - 1
6:     legal ← 0
7:   else
8:     action(step) ← 0
9:     legal ← 1
10:  end if
11:  if packets_left < Network(1) then
12:    legal ← 1
13:  end if
14:  step ← step + 1
15: end while

```

Here, we give an algorithm for distributing the data on a multilane network, running at the BS. The assumptions made in Algorithm 1 hold here as well. The input to Algorithm 2 is a  $n$  by  $m$  matrix  $\text{Network}$  where  $n$  is the number of lines and  $m$  is

the maximum number of nodes per line. It is further assumed that the vector `Est_trans_time` of size  $n$  is initialized with the respective  $T(\nu)$  of each line.

**Algorithm 2:** Determines BS actions in multiline networks

```

input: Network
output: action
1: step ← 1, prev_legal ← ones(1,n),
   legal ← ones(1,n), packets_left ← ∑i,j
   Network(i,j)
2: ∀ packets_left_for_branch(i) ← ∑j
   Network(i,j)
3: while packets_left ≠ 0 do
4:   (y, ind) = max(Est_trans_time.*legal)
5:   if y = 0 then
6:
7:     for i = 1 to nb_of_branches do
8:
9:       if packets_left_for_branch(i) ≠ 0 then
10:        ind = i
11:       end if
12:     end for
13:     action(step) ← 0
14:   else
15:     action(step) ← ind
16:     packets_left ← packets_left - 1
17:     packets_left_for_branch(ind) - 1
18:   end if
19:   legal ← ones(1,nb_of_branches)
20:   for i = 1 to nb_of_branches do
21:
22:     if packets_left_for_branch(i) = 0 then
23:       legal(i) ← 0
24:     end if
25:   end for
26:   tabtest ← ∑ (Network(ind,1:nb_of_
   nodes)) - Network(ind,1)
27:   if (tabtest > 0 and action(step) ≠ 0)
   then
28:
29:     if packets_left_for_branch(ind) ≥
   Network(ind,1) then
30:       legal(ind) ← 0
31:     end if
32:   end if
33:   for i = 1 to nb_of_branches do
34:     if (prev_legal(i) = 1 and i ≠ ind) then
35:       Est_trans_time(i) ← Est_trans_time(i) + 1
36:     end if
37:   end for
38:   prev_legal ← legal
39:   step ← step + 1
40: end while

```

## REFERENCES

- [1] J. R. Agre, L. P. Clare, G. J. Potty, and N. P. Romanov, "Development platform for self-organizing wireless sensor networks," in *Proc. SPIE Unattended Ground Sensor Technologies Applications*, vol. 3713, 1999, pp. 257–268.
- [2] E. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc wireless networks," *IEEE Pers. Commun. Mag.*, pp. 46–55, Apr. 1999.
- [3] V. Bharghavan, A. Demers, S. Shenkar, and L. Zhang, "MACAW: A media access protocol for wireless LAN," in *Proc. SIGCOMM Conf. Communications Architectures, Protocols, Applications*, Aug. 1994, pp. 212–225.
- [4] J. H. Ju and V. O. K. Li, "TDMA scheduling design of multihop packet radio networks based on latin squares," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1345–1352, Aug. 1999.
- [5] M. Pursley, "The role of spread spectrum in packet radio networks," *Proc. IEEE*, pp. 116–134, Jan. 1987.
- [6] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Pers. Commun.*, vol. 7, pp. 16–27, Oct. 2000.
- [7] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. 6th Ann. Int. Conf. Mobile Computing and Networking*, Boston, MA, Aug. 2000, pp. 56–67.
- [8] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, pp. 388–404, Mar. 2000.
- [9] F. T. Leighton, B. M. Maggs, and S. B. Rao, "Packet routing and job-shop scheduling in  $O(\text{Congestion} + \text{Dilation})$  Steps," *Combinatorica*, vol. 14, pp. 167–186, 1994.
- [10] P. Fraigniaud, "Minimum-time broadcast under edge-disjoint paths modes," in *Proc. Int. Conf. Fun with Algorithm*, Italy, 2001, pp. 133–148.
- [11] —, "Approximation algorithms for minimum-time broadcast under the vertex-disjoint paths modes," in *Proc. Eur. Symp. Algorithms*, Denmark, 2001, pp. 440–451.
- [12] B. S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc, and W. Rytter, "Deterministic broadcasting in ad hoc radio networks," *Distrib. Comput.*, pp. 27–38, 2002.
- [13] C. Florens and R. McEliece, "Scheduling algorithms for wireless ad hoc sensor networks," in *Proc. IEEE Global Telecommunications Conf. 2002*, Nov. 2002, pp. 6–10.
- [14] —, "Packets distribution algorithms for sensor networks," in *Proc. IEEE INFOCOM 2003*, Apr. 2003, pp. 1063–1072.



**Cédric Florens** (S'02) received the Diplôme d'ingénieur from the Ecole supérieure d'Ingénieurs en Electronique et Electrotechnique, Paris, France, and the M.S. degree in electrical engineering from the California Institute of Technology, Pasadena, in 1999. He is currently working toward the Ph.D. degree in electrical engineering at the California Institute of Technology.

From 1999 to 2000, he was with the Deep Space Concepts Communication Group, Jet Propulsion Laboratory, California Institute of Technology. From 2000 to 2001, he was employed by QPlus Networks, Long Beach, CA, an optical telecommunications start-up.



**Massimo Franceschetti** (M'98) received the Laurea degree (*magna cum laude*) in computer engineering from the University of Naples, Naples, Italy, in 1997 and the M.Sc. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1999 and 2003, respectively. During his studies, he spent one semester in the Department of Computer Science, University of Edinburgh, Edinburgh, U.K., thanks to a fellowship by the Students Award Association of Scotland.

He is currently a Postdoctoral Scholar at the University of California at Berkeley. His research interests include network models, wave propagation in random media, stochastic systems, and distributed computing.

Dr. Franceschetti is a recipient of the 1999 UPE/IEEE Computer Society Award for Academic Excellence, the 2000 Caltech Walker von-Brimer Foundation Award for Outstanding Research Initiative, and the 2003 Caltech Wilts Prize for best Thesis in electrical engineering.



**Robert J. McEliece** (M'70–SM'81–F'84) was born in Washington, DC, in 1942. He received the B.S. and Ph.D. degrees in mathematics from the California Institute of Technology, Pasadena, in 1964 and 1967, respectively, and attended Trinity College, Cambridge University, Cambridge, U.K., during 1964–1965.

From 1963 to 1978, he was with the Jet Propulsion Laboratory, California Institute of Technology, where he was Supervisor of the Information Processing Group, Communications Research Section, from 1971 to 1978. From 1978 to 1982, he was a Professor of Mathematics and a Research Professor at the Coordinated Science Laboratory, University of Illinois, Urbana–Champaign. Since 1982, he has been on the faculty at the California Institute of Technology, where he is now the Allen E. Puckett Professor of Electrical Engineering. From 1990 to 1999, he served as Executive Officer for Electrical Engineering at the California Institute of Technology. He has been a Consultant in the Communications Research Section of the Jet Propulsion Laboratory since 1978. His research interests include deep-space communication, communication networks, coding theory, and discrete mathematics.

Dr. McEliece is a member of the National Academy of Engineering.