Packets Distribution Algorithms for Sensor Networks

Cédric Florens, Robert McEliece

Department of Electrical Engineering
California Institute of Technology
Pasadena, California, USA
E-mail:{florens, rjm}@systems.caltech.edu

Abstract— **In this paper, we study, via simple discrete mathematical models, the problems of data distribution and data collection in wireless sensor networks. The work that follows continues the work presented by the authors in [1] where the focus was on sensor networks equipped with unidirectional antenna elements. Here we shift our interest to networks equipped with omnidirectional antenna elements. In particular we show how the data distribution and collection tasks can be performed optimally (with respect to time) on tree networks and give the corresponding time performances of those strategies. We also present a strategy for general graph networks that performs within a factor of 3 of the optimal performance. Finally we compare the performance of a network equipped with omnidirectional antenna elements with one equipped with unidirectional antenna elements. We show the latter outperforms the former by 33% at most in tree networks. To that purpose we included relevant results on directional antenna sensor networks, partly obtained in [1].**

I. Introduction

The advancement of Very Large Scale Integration (VLSI) technology has contributed much to the development of micro sensor 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, networks of such devices can autonomously perform various sensing tasks such as environmental (seismic, meteorological) monitoring and military surveillance [2]. These networks are referred to as wireless ad-hoc sensor networks or simply sensor 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 semi permanent stationary configuration is adopted for the purpose of gathering information.

In the area of general ad-hoc networks as well as sensor webs, research has focused on routing [3], medium access control (MAC) [4] [5] and physical layer [6]. [7] and [8] are protocol suites specifically designed for sensor webs. Theoretical results regarding capacity of general static ad-hoc networks first appeared in [9]. Also relevant to our research is the so called packet routing problem which consists in moving packets of data from one location to another as quickly as possible in a network and has been extensively studied in conjunction with wireline and wireless network models (see for example [10], [11], [12] and [13]). To the best of our knowledge however no results specific to sensor networks where, in particular, non uniform data distribution over the network is assumed had yet been derived.

In [1] the authors studied data distribution and data collection in sensor networks equipped with directional antenna elements. This paper extends results obtained in [1] to sensor networks equipped with omnidirectional antenna elements. In addition results concerning the impact of network cycles on the optimality of our algorithms are presented. New results concerning time performance of optimal algorithms in directional sensor networks have also been included for the purpose of comparing unidirectional and omnidirectional systems.

We think of a sensor network as having two main phases of operation (in stationary state, after the nodes have organized themselves into a network). In the first phase or measuring phase, area monitoring results in an accumulation of data at each sensor, 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 of such data transfers.

This paper is organized as follows: In section II we define the model that we have adopted for a sensor network. We present our results for networks equipped with omnidirectional antennas in section III. In section IV, we briefly recap results regarding directional systems and present new ones. We present a comparison analysis of the two types of systems (i.e omnidirectional and unidirectional) in section V and conclude in section VI.

II. Model and problem statement

We define a sensor network as a finite collection of n identical nodes $\{N_1, ..., N_n\}$. Each node N_i is associated with an integer p_i that represents the number of data packets stored at this node at the end of the measuring phase. 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 base station have a common transmission range r. A node (BS included) cannot receive and transmit at the same time. The interference model as defined in [9] 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:

This work was performed at Caltech and supported by The Lee Center for Networking.

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

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 $\{V, E, \mathbf{p}\}$ where $V = \{N_0, ..., N_n\}$, E denotes the set of links and $\mathbf{p} = (p_1, ..., p_n)$. 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 (a necessary condition for that connection to be present is that the distance between the two nodes is less than or equal to the transmission range r). By its nature this link is single duplex bidirectional. Our goal is to route the data contained at each node to the BS as efficiently as possible. We refer to this as the *data collection problem.*

III. Omnidirectional antenna systems

The objective of this section is to construct an optimal strategy for collecting data on a tree network as well as deriving a closed form formula for the time performance of such strategy. The subsections entitled "Line Networks", "Multiline networks", "Tree networks, case where degree of base station is 1" are the building blocks of that strategy. Lastly, based on those results a procedure is proposed to distribute data on general connected graphs and its time performance is bounded.

A. Line Networks

In this subsection we consider a line network (an example of which is given in Fig. 1) of sensor nodes. A BS is placed at one end of the network. The case where the BS is placed anywhere on the line network (i.e. not at one end of the line) may be seen as a 2-line network where two line networks meet at the BS. This case is therefore a particular case of a multiline network and is studied in the next section. We assume sensor nodes are regularly placed along the network. We denote by d the distance between any 2 nodes. Assume each node is equipped with omnidirectional antennas allowing transmissions over a distance r where $d < r < 2d$. Further assume that δ is such that $(1+\delta)r < 2d$. It is straightforward to extend the following results to more general line networks where nodes are randomly placed along a line and to different values of r, δ, as long as end to end connectivity of the network is ensured. Instances of such scenarios are studied in appendix II. Let's denote node N_i by its distance to the BS in number of hops, that is i. We denote by $i \to i+1$ a transmission from node i to node $i+1$. Our goal is to determine the minimal duration of the transfer phase and an associated optimal communication strategy (Note that in general such a strategy is not unique).

For purpose of solving this problem we look initially at the following converse problem (which we shall subsequently refer to as the *distribution problem*); instead of nodes sending their respective data packets to the BS, assume the BS is to transmit data packets to nodes. Our goal, determining minimal duration of transfer phase and an associated optimal communication strategy, remains unchanged. This problem is of separate interest in sensor networks.

We propose the following simple greedy 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 a packet's destination are required to forward that packet as soon as it arrives (that is in the time slot following its arrival). Following is algorithm 1 running at the BS.

Given a line network (represented by the vector $Network = \mathbf{p}$), it dictates the BS actions at each time step: remain idle ($action = 0$) or transmit ($action = 1$). The result is stored in the vector $action$. 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 that 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

```
input: Network
output: action
step ← 1, packts_left1 ← Network(1), packts_left2 ← Network(2),
  packts_left3 ← $\sum_{i\geq 3}$Network(i), packts_left ← $\sum_i$Network(i)
while packts_left $\neq$ 0 do
    while packts_left3 $\neq$ 0 do
        action(step) ← 1
        action(step+1) ← 0
        action(step+2) ← 0
        step=step+3
        packts_left3=packts_left3-1
    end while
    while packts_left2 $\neq$ 0 do
        action(step) ← 1
        action(step+1) ← 0
        step=step+2
        packts_left2=packts_left2-1
    end while
    while packts_left1 $\neq$ 0 do
        action(step) ← 1
        step=step+1
        packts_left1=packts_left1-1
    end while
    packts_left ← packts_left-1
end while
```

The procedure is illustrated in the example of Fig. 1 where $V = \{0,1,2,3,4,5,6,7,8,9\}$, $E = \{(i, i+1), 0 \leq i \leq 8\}$, $\mathbf{p} = (2,1,0,0,0,0,0,1,1)$, $d < r < 2d$, $(1+\delta)r < 2d$.

The schedule of transmissions as determined by algorithm 1 is drawn below the network (upper schedule) for the distribution problem. It is performed in 11 TS.

Next we determine the performance of our algorithm in general. Denote T_i the last busy time slot at node i, $1 \leq i \leq n$ in the execution of our distribution algorithm (In the previous example, we have $T_1 = 10, T_2 = 8, T_3 = 7, T_4 = 8, T_5 = 9, T_6 = 10, T_7 = 11, T_8 = 11, T_9 = 9$). Clearly then our algorithm runs in $\max\limits_{1\leq i\leq n}\{Ti\}$. T_i is a function of the distance to the BS, the number of data packets destined for node i (that is p_i) and

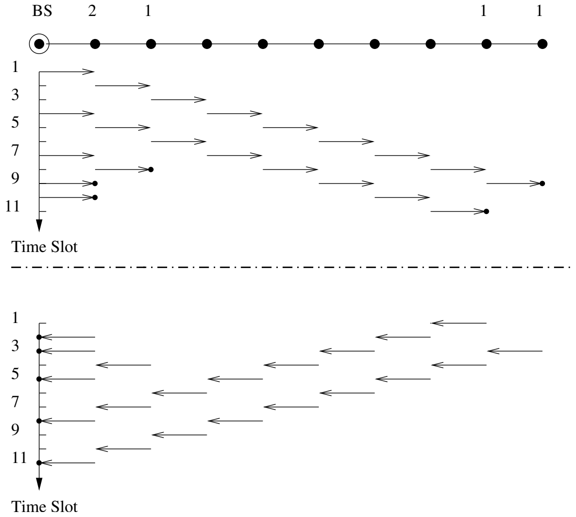


Fig. 1. 10-node line network ($p_1 = 2, p_2 = p_8 = p_9 = 1, p_3 = ... = p_7 = 0$) followed by optimal transmission schedules for the distribution (upper schedule) and collection (lower schedule) problem (omnidirectional antennas assumed). They are symmetric of one another. The job is performed in 11 TS

the number of data packets forwarded by node i.

Lemma 1: Assuming $p_i = 0$ for $i > n$, we have:

$$T_1 = \begin{cases} 3\sum_{j\geq 3} p_j - 1 & \text{if } p_1 = 0, p_2 = 0 \text{ and} \\ & \sum_{j\geq 3} p_j \geq 1 \\ p_1 + 2p_2 + 3\sum_{j\geq 3} p_j & \text{otherwise} \end{cases}$$

$$T_2 = 2p_2 + 3\sum_{j\geq 3} p_j$$

$\forall i \geq 3$

$$T_i = \begin{cases} i - 2 + 3\sum_{j>i} p_j & \text{if } p_i = 0 \text{ and } \sum_{j>i} p_j \geq 1 \\ i + 3\sum_{j>i} p_j & \text{if } p_i = 1 \\ i - 3 + 3\sum_{j\geq i} p_j & \text{if } p_i \geq 2 \end{cases} \tag{2}$$

Proof:

Denote f_i the number of data packets forwarded by node i.

If $i = 1$,

$$p_1 = 0, p_2 = 0, f_i \geq 1 \Rightarrow T_i = 3(f_i - 1) + 2 + (i - 1)$$
$$\text{otherwise}, T_i = p_1 + 2p_2 + 3(f_i - p_2)$$

If $i = 2$,

$$T_i = 2p_2 + 3(f_i - p_2)$$

$\forall i \geq 3$,

$$p_i = 0, f_i \geq 1 \Rightarrow T_i = 3(f_i - 1) + 2 + (i - 1)$$
$$p_1 \geq 1 \Rightarrow T_i = 3f_i + 1 + (i - 1)$$
$$p_i \geq 2 \Rightarrow T_i = 3f_i + 3(p_i - 1) + 1 + (i - 1)$$

but,

$$f_i = \sum_{j>i} p_j$$

hence the stated result. ■

Clearly the maximum of T_i is obtained over the set $\{i \geq 1 \,|\, p_i \neq 0\}$. We define, for a given sensor network, $T_o(\mathbf{p})$ the minimum length of a time schedule over all time schedules that perform the distribution job. Thus we have the following result:

Lemma 2:

$$T_o(\mathbf{p}) \leq \max_{\{i\geq 1 \,|\, p_i \neq 0\}} T_i \tag{3}$$

Let's now derive a lower bound on $T_o(\mathbf{p})$.

Lemma 3: Assuming $p_i = 0$ for $i > n$, we have:

$$T_o(\mathbf{p}) \geq \max_{1\leq i\leq n} (i - 1 + p_i + 2p_{i+1} + 3\sum_{j\geq i+2} p_j) \tag{4}$$

Proof: Consider node $i \geq 1$, assume there exists $k \geq i$ such that $p_k \geq 1$. Then

- edge $(i-1, i)$ is activated $\sum_{j\geq i} p_j$ TS.
- edge $(i, i+1)$-if it exists- is activated $\sum_{j\geq i+1} p_j$ TS.
- edge $(i+1, i+2)$-if it exists- is activated $\sum_{j\geq i+2} p_j$ TS.

Clearly transmissions $i-1 \rightarrow i$, $i \rightarrow i+1$, $i+1 \rightarrow i+2$, $\forall i \geq 1$ may not occur concurrently (channel reuse constraints). Besides from our initial assumptions we know that idle time of nodes $\in \{i, i+1, i+2\} \geq i - 1$. Therefore,

$$T_o(\mathbf{p}) \geq \sum_{j\geq i} p_j + \sum_{j\geq i+1} p_j + \sum_{j\geq i+2} p_j + (i - 1) \triangleq S_i$$

We have $\forall i \quad T_o(\mathbf{p}) \geq S_i$, thus $T_o(\mathbf{p}) \geq \max_i S_i$. ■

Next we prove that the lower bound on $T_o(\mathbf{p})$ derived in lemma 2 equals the upper bound derived in lemma 3 and hence that the proposed schedule is optimal.

Theorem 4: Assuming $p_i = 0$ for $i > n$, we have:

$$T_o(\mathbf{p}) = \max_{1\leq i\leq n} (i - 1 + p_i + 2p_{i+1} + 3\sum_{j\geq i+2} p_j) \tag{5}$$

Proof: Assume there exists j such that $\forall i \neq j$, $T_j \geq T_i$, $T_{j+1} < T_j$

- if $j = 1 \Rightarrow S_1 \geq T_1 \Rightarrow T_1 = S_1$
- if $j = 2 \Rightarrow p_2 \geq 1$, $p_1 = 0$
$\Rightarrow T_2 - S_2 = p_2 + p_3 - 1 \geq 0 \Rightarrow T_2 \geq S_2$
$p_1 = 0 \Rightarrow T_1 = T_2 \Rightarrow S_1 \geq T_1$
- if $j \geq 3 \Rightarrow p_{j-2} = 0$, $p_{j-1} = 0$, $p_j \geq 1$
$S_{j-2} = j - 3 + 3\sum_{i\geq j} p_i$
$p_j = 1 \Rightarrow T_j = S_{j-2}$
$p_j \geq 2 \Rightarrow T_j = S_{j-2}$

■

Corollary 5: In the particular case where no three consecutive components of vector $\mathbf{p}$ equal zero, Theorem 4 reduces to:

$$T_o(\mathbf{p}) = p_1 + 2p_2 + 3\sum_{i\geq 3} p_i \tag{6}$$

We now come back 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 of Fig. 1) provides us with an optimal solution, the time to transmit data packets from nodes to the BS being indeed 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 $T_o(\mathbf{p}) + 1 - j$ 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 case of data collection; the described algorithms don't 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.

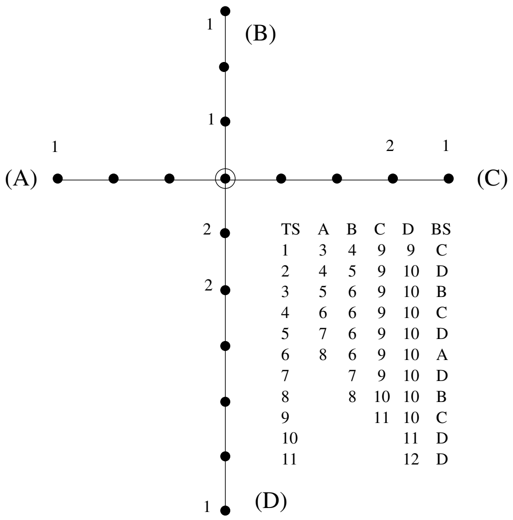
B. Multiline networks

In this section we consider multiline sensor networks, by which we mean multiple line sensor networks meeting in one single point, the BS. Fig. 2 and Fig. 3 are examples of such networks. Next we give an algorithm for distributing the data on a multiline network, running at the BS. The assumptions made in algorithm 1 hold here as well. The input to the procedure is a n by m matrix $Network$ where n is the number of branches (=lines) and m is the maximum number of nodes per branch. It is further assumed that the vector Est_trans_time of size n is initialized with the respective $T(\mathbf{p})$ of each line network.

Algorithm 2 determines BS actions in multiline network

```
input: Network
output: action
 1: step ← 1, prev_legal ← ones(1,n), legal ← ones(1,n)
 2: packts_left ← $\sum_{i,j}$Network(i,j)
 3: $\forall i$ packts_left_for_branch(i)← $\sum_j$Network(i,j)
 4: while packts_left $\neq$ 0 do
 5:   (y,ind)=max(Est_trans_time.*legal)
 6:   if y=0 then
 7:
 8:     for i=1 to nb_of_branches do
 9:
10:       if packts_left_for_branch(i) $\neq$ 0 then
11:         ind=i
12:       end if
13:     end for
14:     action(step) ← 0
15:   else
16:     action(step) ← ind
17:     packts_left ← packts_left-1
18:     packts_left_for_branch(ind) ← packts_left_for_branch(ind)-1
19:   end if
20:   legal← ones(1,nb_of_branches)
21:   for i=1 to nb_of_branches do
22:
23:     if packts_left_for_branch(i)=0 then
24:       legal(i) ← 0
25:     end if
26:   end for
27:   tabtest ← sum(Network(ind,1:nb_of_nodes))-Network(ind,1)
28:   if (tabtest > 0 & action(step) $\neq$ 0) then
29:
30:     if packts_left_for_branch(ind) $\geq$ Network(ind,1) then
31:       legal(ind)← 0
32:     end if
33:   end if
34:   for i=1 to nb_of_branches do
35:     if (prev_legal(i)=1 & i $\neq$ ind) then
36:       Est_trans_time(i) ← Est_trans_time(i)+1
37:     end if
38:   end for
39:   prev_legal ← legal
40:   step ← step+1
41: end while
```

The algorithm running at the BS determines at each TS toward which branch transmit, if transmission is possible at all. The direction of transmission is greedily decided, based on estimates (one estimate per branch) of the completion time of the algorithm. Initial estimate for a given branch is determined by equation (5). These estimates are in fact the earliest possible completion times for each branch. The legal direction associated with the biggest estimate is chosen (a legal transmission is one that respects the channel reuse constraints, so for example it is not legal for our algorithm to transmit in two successive TS toward a given node located at distance greater than 2 from the BS), ties being broken randomly. When no legal direction exists the BS remains idle. After a decision has been made (transmit toward a particular direction or stay idle) the estimates at each branch must be updated: if a legal direction was not chosen, its new estimate is its old estimate $+0/1$ depending on whether the time completion for the particular branch remained unchanged or was increased by one (this should be tested in line 35 of the above algorithm but does not specifically appear for the sake of simplicity) . Illegal direction estimates remain unchanged. The idea is to minimize at each TS the overall estimate of the transmission time by minimizing the completion time of each branch.



TS	A	B	C	D	BS
1	3	4	9	9	C
2	4	5	9	10	D
3	5	6	9	10	B
4	6	6	9	10	C
5	7	6	9	10	D
6	8	6	9	10	A
7		7	9	10	D
8		8	10	10	B
9			11	10	C
10				11	D
11				12	D

Fig. 2. Optimal distribution schedule for BS on a 4-line sensor network. The completion time is 12 TS.

Next we illustrate the procedure on an example (Fig. 2). In the accompanying table, we list data transfer completion time estimates at each TS and the corresponding decision made by the BS (as to which direction to choose). As previously stated the initial completion time estimates are computed using equation (5). The table reads as follows. TS 1: All 4 transmission directions are legal. The BS chooses to transmit toward branch C (it could have chosen D as well, as ties are broken randomly). At TS 2, transmitting toward C is not a legal move, the legal transmission direction associated with the biggest estimate is D (notice that transmitting toward A or B makes the overall completion time estimate be 11 TS, whereas transmitting toward D leaves the completion time estimate unchanged (10 TS), so D is also the legal move that minimizes the estimated completion

time), etc. 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 12TS (an obvious lower bound on the time performance is 11 TS corresponding to 11 data packets).
The previously described algorithm is optimal when the number of data packets at distance 0 and 1 from the BS is zero. If it is not the case, the algorithm needs to be refined, in particular estimates ties should not be broken randomly in general. This as well as a general case proof will appear in a forthcoming paper [14]. In this proof we assumed that relay sensor nodes can only perform simple receive and forward type operations in which a data packet is to be forwarded in the TS following its arrival at a relay node. Note that time performance may be further improved, if we assume that nodes have the ability to perform store and forward type operations (that is store data to be relayed). This was not the case for directional antenna systems. This is illustrated in the following example (Fig. 3: if the simplest relay nodes are being used the completion time is 10TS, whereas it may be as low as 9TS when the smarter nodes are used. However in the directional antenna case the time performance is 9TS either way. This issue is further explored in [14].

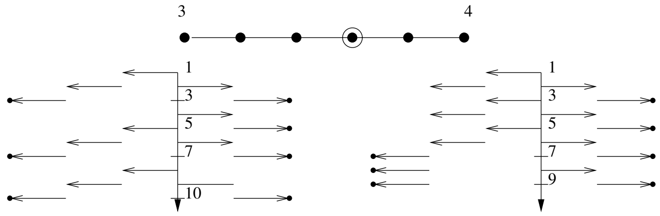


Fig. 3. Optimal distribution schedules of a 2-line sensor network. The completion time are respectively 10 TS (dumb sensor node hypothesis) and 9 TS (smart sensor hypothesis).

C. Tree networks, case where degree of base station is 1

Throughout this paragraph we assume that the degree of the root of the considered graphs is one.

Definition 6: We define the *equivalent linear network* $(G_l, E_l, \mathbf{p_l})$ of a network $(G, E, \mathbf{p})$. If $G = \{N_0, N_1, \ldots, N_n\}$ and $\mathbf{p} = (p_1, \ldots, p_n)$ then $G_l = \{0, 1, \ldots, m \leq n\}$, $E_l = \{(i-1, i), 1 \leq i \leq m\}$ and $\mathbf{p_l} = (p_{l1}, \ldots, p_{lm})$ where $p_{lj} = \sum_{i \,|\, d(N_0,N_i)=j} p_i$
This definition is illustrated in Fig. 4 ($n = 15$, $m = 9$) and Fig. 1 (equivalent line network).

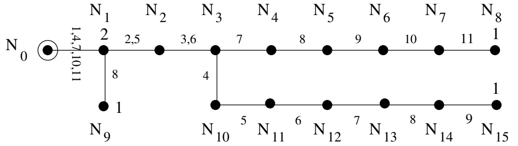


Fig. 4. A 16-node tree network with degree of BS= 1, the equivalent linear network is drawn in Fig. 1. Transmission TS are written next to the edges.

The equivalent linear network's schedule may serve as a schedule for the initial tree network. Next we explain how transmission time slots for $(G_l, E_l, \mathbf{p_l})$ (determined by running algorithm 1) may be mapped onto $(G, E, \mathbf{p})$. 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 *path*(P) from the BS to its destination node where *path*(P) denotes the finite sequence of edges $(e_1, ..., e_k)$ traversed in that order by P. For convenience we shall write $path(P)$ as the sequence of vertices $(vertices(e_1), ..., vertices(e_k))$. We define $\mathfrak{P}_\alpha = \{P \,|\, \exists e \in E_\alpha \cap path(P)\}$. We define $\mathfrak{T}_\alpha = \{\text{TS used by } (\alpha, \alpha+1) \in E_l\}$. We have: $|\mathfrak{P}_\alpha| = \sum_{(N_i,N_j) \in E_\alpha} (p_j + f_j) = \sum_{k>\alpha} p_{lk} = |\mathfrak{T}_\alpha|$. 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 $\mathfrak{P}_\alpha^{(N_{i_0}, N_{j_0})} = \{P \,|\, (N_{i_0}, N_{j_0}) \in path(P)\} \subseteq \mathfrak{P}_\alpha$. (N_{i_0}, N_{j_0}) is associated with time slots $g(\mathfrak{P}_\alpha^{(N_{i_0}, N_{j_0})})$. In the example of Fig. 4, we have: $\{P\} = \{P_1, P_2, \ldots, P_5\}$ where the first packet is characterized by $path(P_1) = (N_0, N_1, N_2, N_3, N_{10}, N_{11}, N_{12}, N_{13}, N_{14}, N_{15})$, the second one by $path(P_2) = (N_0, N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8)$, the third one by $path(P_3) = (N_0, N_1, N_9)$, and finally the fourth and fifth ones by $path(P_4) = path(P_5) = (N_0, N_1)$. We also have $E_1 = \{(N_1, N_2), (N_1, N_9)\}$, $\mathfrak{P}_1 = \{P_1, P_2, P_3\}$, $\mathfrak{T}_1 = \{2,5,8\}$, and $\mathfrak{P}_1^{(N_1,N_2)} = \{P_1, P_2\}$. Thus edge (N_1, N_2) is associated with time slots $g(\mathfrak{P}_1^{(N_1,N_2)}) = \{2,5\}$. Thus algorithm 1 run on the equivalent linear network provides a BS transmission schedule. Intermediate nodes simply forward data packets to further nodes as they arrive (in the TS following their arrival). This requires a routing table at junction nodes.

Although an equivalent linear network has a reduced set of possible concurrent transmissions, this procedure produces an optimal transmission schedule. The following proof is based on the fact that transmissions that can occur in one case and not in the other are not helpful in routing data faster. This is essentially due to the fact that any route from the BS to a leaf necessarily includes link $(0, 1)$ i.e. from the BS to the unique node at distance one from the BS which constitutes a bottleneck.

Lemma 7: Given any tree $\mathcal{T}$ such that degree of BS is one, if $T_o(\mathcal{T})$ denotes the min data distribution time performance, and p_j denotes the number of data packets at distance j from the BS, then:

$$T_o(\mathcal{T}) \geq \max_i (i - 1 + p_i + 2p_{i+1} + 3 \sum_{j>i+1} p_j) \qquad (7)$$

Proof: Edges at distance i from the BS are activated $\sum_{j \geq i} p_j$ times, edges at distance $i+1$ from the BS are activated $\sum_{j \geq i+1} p_j$ times and edges at distance $i+2$ from the BS are activated $\sum_{j \geq i+2} p_j$ times. In a given TS, the distance (to the BS) difference of any two data packets in transit is at least 3 hops. This implies in particular that no 2 edges whose distance difference to the BS is less than or equal to 2 hops may be activated simultaneously. ■

In this proof we assumed that relay sensor nodes can only perform simple receive and forward type operations in which a data packet is to be forwarded in the TS following its arrival at a relay node. Note that time performance may be further improved, if we assume that nodes have the ability to perform store and forward type operations (that is store data to be relayed). This, again, was not the case for directional antenna systems. This is illustrated in the following example: if the simplest relay nodes are being used, $T_o(\mathcal{T}) = 6$ TS, whereas $T_o(\mathcal{T}) = 5$ TS may be obtained with the schedule: TS 1: $N_0 \to N_1$, TS 2: $N_1 \to N_2$, TS 3: $N_0 \to N_1$, TS 4: $N_1 \to N_3$, TS 5: $N_1 \to N_2$, $N_3 \to N_4$. However in the directional antenna case $T_u(\mathcal{T}) = 5$ TS either way. This issue is further explored in [14].

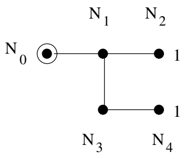


Fig. 5. 6-nodes sensor network, dumb sensor nodes case $T_o(\mathcal{T}) = 6$ TS, smart sensor nodes case $T_o(\mathcal{T}) = 5$ TS

D. Tree Networks

The procedures described in the previous sections may be combined into a strategy for data distribution/collection on tree networks as follows:

1) linearize the subtrees attached to the BS according to the procedure described in III. C
2) apply multiline distribution algorithm to the resulting multiline system as described in III. B

One can show from III. A, B and C that this procedure is optimal on general tree networks. In the following theorem we give without proof a closed form expression for its time performance. A manuscript detailing the missing proofs is in preparation [14].

Time performance on tree networks:

For purpose of deriving the time performance of our strategy on tree networks, we start by defining the equivalent network N_e of a multiline network N in the following manner: To each branch (= line) B_k of N and associated data vector $\mathbf{p^k}$ corresponds a branch B'_k in N_e and associated data vector $\mathbf{p'^k}$ such that:

$$\begin{aligned}
&i = 1 && p'^k_1 = p^k_1. \\
&i = 2 && p'^k_{T_o(\mathbf{p}^k) - T^k_i + 2j + i} = 1 \\
& && \text{for} \quad 0 \le j \le l - 1 \quad \text{if} \quad p^k_i = l \ge 1. \\
&i \ge 3 && p'^k_{T_o(\mathbf{p}^k) - T^k_i + 3j + i} = 1 \\
& && \text{for} \quad 0 \le j \le l - 1 \quad \text{if} \quad p^k_i = l \ge 1. \\
& && p'^k_i = 0 \quad \text{otherwise.}
\end{aligned} \tag{8}$$

Theorem 8: If $\mathcal{T}$ is a tree and $T_o(\mathcal{T})$ denotes the minimum distribution time over $\mathcal{T}$. If p^k_j denotes the number of data packets at distance j from the BS along branch k, then, if $p_0 = p_1 = 0$:

$$T_o(\mathcal{T}) = \max_i \, (i - 1 + \sum_{j \ge i} p'_j) \tag{9}$$

where $p'_j = \sum_k p'^k_j$ and p'^k_j is obtained from p^k_j by equation (8).

Proof: This follows from results in appendix I and the proof of optimality of the strategy on multiline networks, manuscript in preparation [14]. ■

E. General connected sensor networks

For purpose of analyzing the time performance of data distribution algorithms on general sensor networks we denote by $\mathcal{T}_{SP}(G)$ a shortest path spanning tree of the underlying network graph G. Note that one can show that shortest path spanning trees always exist by using Dijkstra algorithm. Such a tree may not be unique. The following theorem provides a motivation for choosing a shortest path spanning tree.

Theorem 9: $\forall \mathcal{T}$, a spanning tree of G

$$T_o(\mathcal{T}_{SP}) \le T_o(\mathcal{T}) \tag{10}$$

The presence of cycles in a network G will affect the optimal time performance of distributions algorithms as compared with the optimal time performance over $\mathcal{T}_{SP}(G)$. Subsequently we attempt to quantify this phenomenon as well as giving some simple procedures to distribute data over G.

First we note that cycles may help or hurt the time performance of the optimal scheduling strategy on omnidirectional systems (in contrast with directional systems). That is $T_o(G)$ may be larger or smaller than $T_o(\mathcal{T_{SP}})$as shown in the examples of Fig. 6 and Fig. 7.

Theorem 10: For any (connected) graph G, and any shortest path spanning tree $\mathcal{T}_{SP}$

$$\frac{T_o(\mathcal{T}_{SP})}{3} \le T_o(G) \tag{11}$$

Proof: define: $t_1(G)$ the minimum distribution time when transmission and reception are simultaneously allowed in a TS at any given node. Clearly $t_1(G) \le T_o(G)$. By corollary 21 (appendix I) we also have: $t_1(G) = t_1(\mathcal{T}_{SP})$. Besides for any connected graph A the following inequality holds: $T_o(A) \le 3t_1(A)$. Choose $A = \mathcal{T}_{SP}$, the inequality follows. ■

Let us next give an example where the lower bound is achieved. Consider a network G where n data packets are stored at distance k hops from the BS in node x. Further assume there are three distinct paths of length k from x to BS (see Fig. 6 where $n = 5$, $k = 6$).

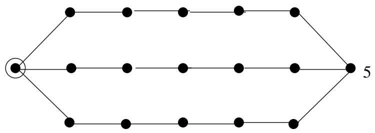


Fig. 6. Example of a network with cycles. We have $T_o(G) = 10$ TS, $T_o(\mathcal{T}_{SP}) = 18$ TS

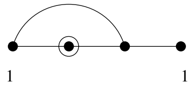


Fig. 7. Example of a network with cycles. We have $T_o(G) = 3$ TS, $T_o(\mathcal{T}_{SP}) = 2$ TS

For all pratical purposes, T_{SP} is the line network $\mathbf{p} = (\overbrace{0, 0, ..., 0}^{k-1}, n)$. We have $T_o(G) = n + k - 1$ (for $k \geq 1$) and $T_o(\mathcal{T}_{SP}) = 3n + k - 3$ (for $k \geq 3$), thus $T_o(G)$ converges toward $T_o(\mathcal{T}_{SP})/3$ when n goes to infinity (for $k \geq 3$).

Strategy and Time performance:

A mere generalization of the strategy developed in IV. C for directional antenna systems, based on extracting a shortest path spanning tree of the sensor network, is not envisageable here, as such as an operation is not *physically* possible when nodes are equipped with omnidirectional antennas. We propose to transmit each data packet to its destination along any shortest path between the BS and its destination. An intermediate node will forward a data packet in the TS following its arrival along that path. Furthest nodes being served first. This is slightly different from algorithm 1 as stated in III. A in the fact that the BS is not to transmit as fast as possible but according to the rule: if previous destination node is at distance greater or equal 3, stay idle 2 TS before sending next packet. If previous destination node is at distance 2 from the BS, stay idle 1 TS before sending new packet. If previous packet is at distance 1, send next packet. The time performance of that strategy is clearly $\max_i (i - 1 + p_i + 2p_{i+1} + 3\sum_{j \geq i+2} p_j)$. However a proof that this strategy may be implemented is required at this point.

Proof: All that is needed is a proof that given any network G equipped with omnidirectional antenna nodes, transmissions originating at any node N_1, at distance i from the BS and at any node N_2, at distance $i + 3$ from the BS may occur concurrently. Note that if node N_1 can reach node N_1' and $d(N_1) = i$ then $d(N_1') \leq d(N_1) + 1$ and $d(N_1) \leq d(N_1') + 1$, therefore $d(N_1') \in \{i - 1, i, i + 1\}$. Assume N_1 attempts to communicate with some node N_1' while N_2 attempts to communicate with node N_2'. One of the attempted communications fails if either there is an edge connecting N_1' and N_2 or there is an edge connecting N_1 and N_2'. If $(N_1', N_2) \in E_G$ then $d(N_2) = d(N_1') + 1 \in \{i, i+1, i+2\} < i+3$ which contradicts our hypothesis. If $(N_1, N_2') \in E_G$ then $d(N_2) = d(N_2') + 1 \in \{i, i + 1, i + 2\} < i + 3$ which contradicts our hypothesis. ■

Corollary 11: If p_j denotes the total number of data packets at distance j from the BS,

$$\max_i (i - 1 + \sum_{j \geq i} p_j) \leq T_o(G) \leq \max_i (i - 1 + p_i + 2p_{i+1} + 3 \sum_{j \geq i+2} p_j) \quad (12)$$

The lower bound on $T_o(G)$ is achievable. Indeed in the previously considered example $\max_i (i - 1 + \sum_{j \geq i} p_j) = n + k - 1$. The figure below shows an example where the upperbound is achieved.

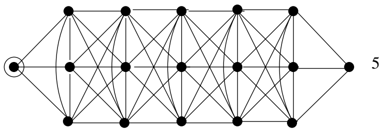


Fig. 8. Example of a network with cycles. We have $T_o(G) = T_o(\mathcal{T}_{SP}) = 18$ TS

In general the upperbound is achieved when any node at distance i from the BS is connected to all the nodes at distance $j \in \{i - 1, i, i + 1\}$.

IV. Directional antenna systems

In this section we list a few relevant results relative to directional antenna systems. Results in part A were proven in [1]. Missing proofs to part B results will be included in a forthcoming paper.

A. Line networks

The strategy to distribute/collect data on a line network $\mathbf{p}$ is described in [1]. We shall not include it here, suffice it to say that it is similar in spirit to algorithm 1 in section III. We shall refer to it as algorithm 1' subsequently. Notations and assumptions introduced in sections II and III are kept. If $T_u(\mathbf{p})$ denotes the minimum length of a time schedule over all time schedules that perform the distribution job (the subscript *u* stands for unidirectional) then we have the following theorem:

Theorem 12:

$$T_u(\mathbf{p}) = \max_{1 \leq i \leq n} (i - 1 + p_i + 2 \sum_{j \geq i+1} p_j) \quad (13)$$

B. Tree networks

The strategy to distribute data on a general tree network rests on two subprocedures:

1) linearize the subtrees attached to the BS
2) distribute data on resulting multiline algorithm

Both procedures are illustrated in [1]. We detail here the time performance of that strategy while omitting any proof of optimality.

Let us first define the *equivalent network* N_e of a multiline network N: To each branch (= line) of N, say B_k, if p_i^k denotes the number of data packets at distance i from the BS along B_k and T_i^k is the last busy time slot in the execution of algorithm 1' for that branch, associate a branch in N_e, say B_k' such that if $p_i'^k$ denotes the number of data packets at distance i from the BS along B_k':

$$\begin{aligned} &i = 1 \quad p_1'^k = p_1^k \qquad (14)\\ &i \geq 2 \quad p'^k_{T_u(\mathbf{p}^k) - T_i^k + 2j + i} = 1 \\ &\qquad\qquad \text{for} \quad 0 \leq j \leq l - 1 \quad \text{if} \quad p_i^k = l \geq 1 \\ &\qquad\quad p_i'^k = 0 \quad \text{otherwise} \end{aligned}$$

Denote by $T_u(\mathcal{T})$ the minimum data distribution time over a tree $\mathcal{T}$, then we have:

Theorem 13: If p_j^k denotes the number of data packets at distance j from the BS along branch k, then:

$$T_u(\mathcal{T}) = \max_i (i - 1 + \sum_{j \geq i} p_j') \quad (15)$$

where $p_j' = \sum_k p_j'^k$ and $p_j'^k$ is obtained from p_j^k by equation (14).

C. Networks with cycles

We propose a data distribution/collection strategy on general graphs. However that 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 and not just any tree.

Theorem 14: $\forall \mathcal{T},$ a spanning tree of G

$$T_u(\mathcal{T}_{SP}) \leq T_u(\mathcal{T}) \tag{16}$$

Theorem 15: For any (connected) graph G, and any shortest path spanning tree $\mathcal{T}_{SP}$ we have:

$$\frac{T_u(\mathcal{T}_{SP})}{2} \leq T_u(G) \leq T_u(\mathcal{T}_{SP}) \tag{17}$$

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 20 (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 $\mathbf{p} = (\overbrace{0, 0, ..., 0}^{k-1}, n)$. 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 16:

$$\max_i (i - 1 + \sum_{j \geq i} p_j) \leq T_u(G) \leq \max_i (i - 1 + p_i + 2 \sum_{j \geq i+1} p_j) \tag{18}$$

Proof: we have from corollary 21 (Appendix I) $t_1(G) = \max\limits_i (i - 1 + \sum_{j \geq i} p_j)$ ■

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

V. Comparison between omnidirectional and directional systems

In order to get a better intuition on how the two systems perform relative to one another, we give the following comparative result for tree networks:

Theorem 17:

$$\forall \mathcal{T} \text{ a tree network,} \qquad 1 \leq \frac{T_o(\mathcal{T})}{T_u(\mathcal{T})} < 1.5 \tag{19}$$

Proof: see [14]. ■

VI. Conclusions and future work

We have proposed optimal strategies to distribute and collect data packets from a tree-like sensor network. The exact performance times of such strategies have been derived. We assessed those strategies on general graph networks. Finally we compared the performance of omnidirectional systems to directional ones.

We are currently working on extending our comparison analysis between directional and omnidirectional systems.

Appendix I
Preliminary Results

In the following section we assume that a network equipped with directional nodes may receive *and* transmit a data packet during any given time step (whereas so far we had assumed that it was only possible to receive *or* transmit a data packet in a given time step). 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 on such networks 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 trough a series of successive building steps.

A. Lower bound on the time performance of data distribution algorithms

Lemma 18: Given any connected graph G, if $t_1(G)$ denotes the time performance of a given data distribution algorithm, and p_j denotes the number of data packets at distance j from the BS, then:

$$t_1(G) \geq \max_i (i - 1 + \sum_{j \geq i} p_j) \tag{20}$$

Proof: $\sum_{j \geq 1} p_j$ 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: $t_1(G) \geq \sum_{j \geq 1} p_j$.

$\sum_{j \geq i} p_j$ data packets must be delivered to nodes at distance greater than $i > 1$. After $\sum_{j \geq i} p_j$ 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: $t_1(G) \geq \sum_{j \geq 1} p_j + i - 1$. 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 time slot following their arrival).

This algorithm is illustrated by an example on Fig. 9.

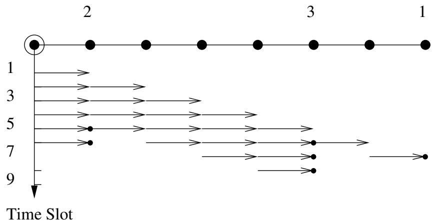


Fig. 9. Optimal distribution schedule for BS on a line network equipped with directional antennas and ability to receive and transmit in the same TS. The completion time is 8 TS.

Proof of optimality and time performance:
Denote T_i the last busy time slot at node i in the execution of our algorithm. Clearly then our algorithm runs in $\max_{1 \leq i \leq n} \{Ti\}$. 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 19:

$$T_i = \begin{cases} i + \sum_{j>i} p_j & \text{if } p_i \leq 1 \\ i - 1 + \sum_{j \geq i} p_j & \text{if } p_i \geq 1 \end{cases} \tag{21}$$

Proof:

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

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

$$f_i = \text{number of pkts forwarded by } i = \sum_{j>i} p_j$$

■

Lemma 20: define: $S_i = \sum_{j \geq 1} p_j + i - 1$, 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 $p_i \geq 1$. Since clearly $\max_i T_i$ occurs in i such that $p_i \geq 1$, 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 16 and p_j is the number of data packets at distance j from the BS. The next corollary follows from lemma 17:

Corollary 21:

$$t_1(G) = \max_{1 \leq i \leq n} (i - 1 + \sum_{j \geq i} p_j) \tag{22}$$

Corollary 22:

$$\forall \mathcal{T} \text{a spanning tree of } G, \quad t_1(\mathcal{T}_{SP}) \leq t_1(\mathcal{T}) \tag{23}$$

APPENDIX II
TOWARD MORE GENERAL LINE NETWORKS

Fig. 10 illustrates a generalized version of the line network described in section III. A. It consists of n randomly located sensor nodes $N_1, \ldots, N_n$ 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 $r \geq \max_{i \geq 0} d(N_i, N_{i+1})$ (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 that one neighbor to the right (resp. left) -those numbers varying from one node to the other. However this appendix focuses on the case $r = 1$ hop and $\delta = m - 1$, fixing the number of neighbors on each side of a node to a constant. The model also assumes that interference occurs over m times the transmission range (Note that in part III, m was was chosen to be 1) . In practice m is often comprised between 2 and 3. For results about more general scenarios, the reader is refered to [14].

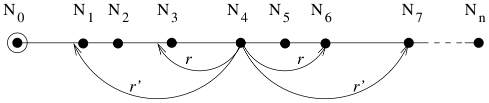


Fig. 10. $(n + 1)$-node line network where $r' = 2r$

A. Directional antenna systems

Strategy for the BS:
Transmit p_n data packets to node N_n first, then p_{n-1} packets to N_{n-1}, and so on, as fast as possible while respecting the channel reuse/ transceiver constraints.

Time performance of strategy:
$= \max_i T_i$ where:

$$T_1 = \begin{cases} \sum_{1 \leq j \leq m-1} j p_j + m \sum_{j \geq m} p_j \text{ if } p_1 \geq 1 \\ 2 + m(\sum_{j \geq m} p_j - 1) \text{ if} \\ \qquad \exists k \geq m \text{ s. t. } p_1 = \ldots = p_{k-1} = 0, p_k \geq 1 \\ 2 + k(p_k - 1) + \sum_{k+1 \leq j \leq m-1} j p_j + m \sum_{j \geq m} p_j \text{ if} \\ \qquad \exists k \, 2 \leq k < m \text{ s. t. } p_1 = \ldots = p_{k-1} = 0, p_k \geq 1 \end{cases}$$

$$T_2 = \begin{cases} \sum_{2 \leq j \leq m-1} j p_j + m \sum_{j \geq m} p_j \text{ if } p_2 \geq 1 \\ 3 + m(\sum_{j \geq m} p_j - 1) \text{ if} \\ \quad \exists k \geq m \text{ s. t. } p_2 = \ldots = p_{k-1} = 0, p_k \geq 1 \\ 3 + k(p_k - 1) + \sum_{k+1 \leq j \leq m-1} j p_j + m \sum_{j \geq m} p_j \text{ if} \\ \quad \exists k \, 2 \leq k < m \text{ s. t. } p_2 = \ldots = p_{k-1} = 0, p_k \geq 1 \end{cases}$$

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

$$\text{if } k \geq 0 \text{ then, } T_{m+k} = \begin{cases} k + m \sum_{j \geq m+k} \text{ if } p_{m+k} \geq 1 \\ k + 1 + m \sum_{j \geq m+k} \text{ if } p_{m+k} = 0 \end{cases}$$

If $T_u^m(\mathbf{p})$ denotes the optimal time performance achievable, then we have $T_u^m(\mathbf{p}) \leq \max_i T_i$. A lower bound on the minimum time performance can be derived as well:

$$T_u^m(\mathbf{p}) \geq$$
$$i - 1 + \sum_{j \geq i} p_j + \sum_{j \geq i+1} p_j + \ldots + \sum_{j \geq i+m-1} p_j \quad \forall i \geq 1$$

Indeed transmissions $i-1 \rightarrow i, i \rightarrow i+1, \ldots, i+m-2 \rightarrow i+m-1$ may not occur concurrently due to channel reuse constraints.
The inequality may be rewritten:

$$T_u^m(\mathbf{p}) \geq$$
$$\max_i (i-1+\sum_{i\leq j\leq i+m-2}(j-i+1)p_j+m\sum_{j\geq i+m-1}p_j) \quad \forall m \geq 2$$

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 $\forall i \neq j_0, T_{j_0} \geq T_i \quad T_{j_0+1} < T_{j_0}$

- if $j_0 = 1$ then $p_1 \geq 1$
  $\Rightarrow S_1 = T_1$
- if $j_0 = 2$ then $p_2 \geq 1, p_1 = 0$
  $\Rightarrow T_2 - S_2 = \sum_{2\leq j\leq m} p_j - 1 \geq 0$
  $p_1 = 0 \Rightarrow T_1 = T_2 \Rightarrow S_1 \geq T_2$
  indeed $S_1 \geq T_1$
  Proof: $p_1 \geq 1 \Rightarrow S_1 = T_1$ and

$$p_1 = 0 \Rightarrow$$
$$S_1 - T_1 = \begin{cases} \sum_{1\leq j\leq m-1} jp_j + m - 2 \geq 0 & (m \geq 2) \text{ or} \\ \sum_{1\leq j\leq k-1} jp_j + k - 2 \geq 0 & (k \geq 2) \end{cases}$$

■

- if $2 < j_0 < m \Rightarrow p_i \geq 1, p_1 = \ldots = p_{i-1} = 0$
  $\Rightarrow T_{j_0} - S_1 = -\sum_{1\leq j\leq i-1} jp_j = 0$

- if $j_0 = m+k \quad k \geq 0 \Rightarrow p_{m+k} \geq 1 \quad p_k = \ldots = p_{k+m-1} = 0$
  $\Rightarrow T_{m+k} - S_{k+1} = -\sum_{k+1\leq j\leq k+m-1}(j-k)p_j = 0$

Therefore $\max_i T_i = \max_i S_i$ and Theorem 23 follows:

Theorem 23:

$$T_u^m(\mathbf{p}) =$$
$$\begin{cases} \max_i (i-1+\sum_{i\leq j\leq i+m-2}(j-i+1)p_j + m\sum_{j\geq i+m-1} p_j) \\ \quad \forall m \geq 2 \\ \max_i (i-1+p_i+2\sum_{j\geq i+1} p_j) \\ \quad \text{if } m = 1 \end{cases}$$

B. Omnidirectional antenna systems

From the previous section ($m \leftarrow m+2$), it follows:

Theorem 24:

$$\forall m \geq 1 \quad T_o^m(\mathbf{p}) =$$
$$\max_i (i-1+\sum_{i\leq j\leq i+m}(j-i+1)p_j + (m+2)\sum_{j\geq i+m+1} p_j)$$

C. Comparison between omnidirectional and directional systems

Theorem 25:

$$1 \leq \frac{T_o^m(\mathbf{p})}{T_u^m(\mathbf{p})} < 1 + \frac{2}{m} \qquad \forall m \geq 2$$

Proof: Assume there exists j_0, $1 \leq j_0 \leq n$ such that for all i, $i \neq j_0 \quad T_{j_0} \geq T_i \,\&\, T_{j_0+1} < T_0 j$

- case: $j_0 = m+2+k \Rightarrow T_{j_0}^o = S_{k+1}$, $T_{j_0}^u = S_{k+3}$
  $\Rightarrow \frac{T_o^m(\mathbf{p})}{T_u^m(\mathbf{p})} = \frac{k+\sum_{j=k+1}^{k+1+m}(j-k)p_j+(m+2)\sum_{j\geq k+m+2} p_j}{k+2+\sum_{j=k+3}^{k-1+m}(j-k-2)p_j+m\sum_{j\geq k+m+2} p_j}$
  $j_0 = m+2+k \Rightarrow p_{k+1} = \ldots = p_{k+1+m} = 0$
  $\Rightarrow \frac{T_o^m(\mathbf{p})}{T_u^m(\mathbf{p})} = \frac{k+(m+2)\sum_{j\geq k+m+2} p_j}{k+2+m\sum_{j\geq k+m+2} p_j} < \frac{m+2}{m}$
- case: $j_0 = m+1 \Rightarrow T_{j_0}^o = S_1$, $T_{j_0}^u = S_2$
  $\Rightarrow \frac{T_o^m(\mathbf{p})}{T_u^m(\mathbf{p})} = \frac{\sum_{j=1}^{m+1} jp_j+(m+2)\sum_{j\geq m+2} p_j}{1+\sum_{j=2}^{m}(j-1)p_j+m\sum_{j\geq m+1} p_j}$
  $j_0 = m+1 \Rightarrow p_1 = \ldots = p_m = 0$
  $\Rightarrow \frac{T_o^m(\mathbf{p})}{T_u^m(\mathbf{p})} = \frac{(m+1)p_{m+1}+(m+2)\sum_{j\geq m+2} p_j}{1+mp_{m+1}+m\sum_{j\geq m+2} p_j} < \frac{m+2}{m}$
- case: $1 \leq j_0 < m \Rightarrow T_{j_0}^o = S_1$, $T_{j_0}^u = S_1$
  $\Rightarrow \frac{T_o^m(\mathbf{p})}{T_u^m(\mathbf{p})} = \frac{\sum_{j=1}^{m+1} jp_j+(m+2)\sum_{j\geq m+2} p_j}{\sum_{j=1}^{m-1} jp_j+m\sum_{j\geq m} p_j} < \frac{m+2}{m}$

■

Note: Bounds in Theorem 25 are tight. This is clear in the case of the lowerbound. As for the upperbound, consider $\mathbf{p} = \mathbf{1}_n$, then we have: $\frac{T_o^m(\mathbf{p})}{T_u^m(\mathbf{p})} = \frac{(\sum_1^{m+1} j+(m+2)\sum_{m+2}^n 1}{\sum_1^{m-1} j+m\sum_m^n 1} = \frac{(m+2)(m+1)/2+(m+2)(n-m-1)}{m(m-1)/2+m(n-m+1)} \underset{n}{\longrightarrow} \frac{m+2}{m}$

References

[1] C. Florens and R. McEliece, "Scheduling algorithms for wireless ad-hoc sensor networks.," *Proceedings of the IEEE Global Telecommunications Conference 2002*, Nov. 2002.

[2] J. R. Agre, L. P. Clare, G. J. Potty, and N. P. Romanov, "Development platform for self-organizing wireless sensor networks," *Proceedings of SPIE, unattended ground sensor technologies and applications*, vol. 3713, pp. 257–268, 1999.

[3] E.M. Royer and C.-K. Toh, "A review of current routing protocols for ad-hoc wireless networks," *IEEE Personal Communications Magazine*, pp. 46–55, Apr. 1999.

[4] V. Bharghavan, A. Demers, S. Shenkar, and L. Zhang, "MACAW: a media access protocol for wireless LAN.," *Proceedings of the SIGCOMM 94 Conference of Communications architectures, protocols and applications*, pp. 212–225, Aug. 1994.

[5] J. H. Ju and V. O. K. Li, "TDMA scheduling design of multihop packet radio networks based on latin squares," *IEEE journal on selected areas in communications*, vol. 17, no. 8, pp. 1345–1352, Aug. 1999.

[6] M. Pursley, "The role of spread spectrum in packet radio networks.," *Proceedings of the IEEE*, Jan. 1987.

[7] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, Oct. 2000.

[8] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, Aug. 2000.

[9] P. Gupta and P.R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, Mar. 2000.

[10] F. T. Leighton, B. M. Maggs, and S. B. Rao, "Packet routing and job-shop scheduling in 0(congestion+dilation) steps," *Combinatorica*, vol. 14, pp. 167–186, 1994.

[11] P. Fraigniaud, "Minimum-time broadcast under edge-disjoint paths modes," *International conference on fun with algorithm*, 2001.

[12] P. Fraigniaud, "Approximation algorithms for minimum-time broadcast under the vertex-disjoint paths modes," *European symposium on algorithms*, 2001.

[13] B. S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc, and W. Rytter, "Deterministic broadcasting in ad hoc radio networks," *Distributed computing*, 2002.

[14] C. Florens and M. Franceschetti, "Data distribution strategies for sensor networks," *manuscript in preparation*.