Improved Lower Bounds for Data-Gathering Time in Sensor Networks

Yoram Revah and Michael Segal
Communication Systems Engineering Department
Ben-Gurion University of the Negev,
Beer-Sheva, Israel 84105

Abstract

Many-to-one packet routing and scheduling are fundamental operations of sensor networks. It is well known that many sensor network applications rely on data collection from the nodes (the sensors) by a central processing device. There is a wide range of data gathering applications like: target and hazard detection, environmental monitoring, battlefield surveillance, etc. Consequently, efficient data collection solutions are needed to improve the performance of the network. In this paper, we assume a known distribution of sources (each node wants to transmit at most one packet) and one common destination (called base station). We provide *via simple mathematical models*, a transmission schedule for routing all the messages to the base station, jointly minimizing both the completion time and the average packet delivery time. We present improved lower bounds for linear, two-branch, and star (or multi-branch) network topologies. All our algorithms run in polynomial time.

1 Introduction

The advancement of Very Large Scale Integration (VLSI) and Micro-Electro-Mechanical Systems (MEMS) technology combined with low power, low cost, digital signal processors (DSPs) and radio frequency (RF) circuits have contributed a lot 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 [1]. These networks are referred to as wireless ad-hoc sensor networks or simply sensor networks. It can also be a collection of mobile sensor nodes that dynamically form a temporary network without the use of any existing network infrastructure or centralized administration.

An important problem in radio networks is scheduling the forwarding of information gathered by sensor nodes. The scheduling process is intended to prevent collisions that might arise from improper or inefficient use of the network resources by random messaging across the network without taking into account the network model. In this work, we aim to solve *via simple mathematical models*, the following problem: Given a certain topology of a radio network and a network model, initial information (messages) located at some nodes and a single designated destination, we analyze and find *(our target function)* an optimal scheduling solution such for all the messages to be routed to the destination in a minimum completion time as well as a minimum average delivery time. We present polynomial time solutions for our problem for three network topologies: Linear, Two-branch, and Star (or multi-branch) network.

Our research can be practically implemented in these networks: for example, whenever a node has a packet to transmit, it sends a very short message (save battery energy) called a “Schedule Request” to a central computer that serves as the only destination in the network. This computer is called a *Base Station* (BS). The requests can be sent over an upstream control channel.

Our problem has been partially addressed over the past few decades. A number of works (see [2–18]) discuss radio networks under a similar network model, but with a different target function that leads to maximizing the number of transmissions in one hop without referring to specific sources and destinations across the networks. This problem and its variations are known to be NP-hard. Other works (see, e.g. [19–27]) considered our problem but under other (weaker) network models. For example, [19] uses the same target function as we suggest, but it discusses several variations of “hot potato” routing. In this model each node can successfully receive and transmit more than one message simultaneously. Sridhran and Krishnamachari [28] deal with some problem of converge-casting flows with rate control from nodes to the root of the given routing tree of the network. Lau and Zhang [29] and Krumme et al. [30] also study the gossiping problem in a two-dimensional grid network topology. They have suggested that the gossiping problem can be studied under four different communication models, which have different restrictions on the use of the links, as well as the ability of a node in handling its incident links. The four models being considered are: (1) the full-duplex, all port model, (2) the full-duplex, one-port model, (3) the half-duplex, all-port model, and (4) the half-duplex, one port model, which can be identified by the labels F*, F1, H*, and H1 respectively.

We assume the network model denoted H1 [29,30] or called “The half duplex one port model”, since this model of communication makes the weakest assumptions about both hardware and software capabilities. Gronkvist [31] assumes a stochastic model for the general network topology problem and presents a number of results under this model. Finally, Florens and McEliece [32] consider exactly our problem under a criterion of minimum completion time, ignoring the requirement of minimizing average delivery time. In fact, their scheduling strategy does not take into account the idle time of the messages and produces unnecessary dependences among messages. This, consequently, causes unnecessary delays for messages. For example, it is unreasonable to not transmit a message toward the destination if it can be transmitted without any delay. They [32] also do not provide any time-complexity analysis of their algorithms. On the other hand, our algorithms' results can serve as new lower bounds taking into account the both criteria: minimizing the time completion and average delivery time.

It should be noted that the presented (optimal) data gathering algorithms are centralized and require cooperation between nodes which is not necessarily compatible with the requirements of sensor networks. Therefore for stronger requirements, these algorithms may no longer be practical. However, they continue to provide a lower bound on data gathering time of any given collection schedule. We focused our analysis on systems equipped with directional antenna since from comparison results (with respect to completion time) between directional antenna systems to omni-directional antenna systems obtained by Florens and McEliece [32] it follows that former outperforms the later by 50% on Linear Network. The idea of using directional antenna in wireless communication is not new. It has been already extensively used in base station of cellular networks for frequency reuse, to reduce interference, and to increase the capacity of allowable users within a cell [33]. However, the applications of directional antenna to wireless ad-hoc or sensor network to reduce the transmit power of each node to achieve power-efficiency in routing problem is relatively new.

This paper is organized as follows: First we explain l the network and channel model with a precise definition of our problem. Next, we address the Linear Network case. After that, we generalize our results to work for two-branch networks. The optimal scheduling strategy under both target functions for star network is explained in Section 3.3. Finally we conclude the paper with directions for further research.

2 Network and Channel Model

A sensor or ad-hoc network is modeled as a graph *G(V,E)* with N nodes $\{v_0, v_1 ....v_N\}$, where each node v_i is a sensor that can transmit and receive data. There is an edge *(v_i,v_j)* iff v_j can hear v_i's transmissions when v_i points its directional transmission antenna towards v_j. We assume that at time t=t_0, each node v_i has at most one message to transmit to the destination. This is referred to as a *legal input*. The network has a special node v_0, referred to as the *Base Station (BS)*, which is the destination of all messages. We also assume that every node in the network including the BS has the same transmission range r and that a node can not transmit and receive message at the same time. In principle, any node can keep an arbitrary number of messages, however as we will see later in our problems it is enough to assume that the capacity of each node's buffer is one message. We assume that all the information about the input and topology of the network is available at the BS and there are separate, collision free, control channels between the BS and the other nodes. We also assume the use of directional antennas. The signal from node v_i to node v_j propagates in a straight line in the direction of node v_j without dispersing to other directions. Based on those assumptions, the conditions for a successful transmission are: A message from node v_i that is transmitted to node v_j, $i, j \geq 0$, arrives successfully to node v_j if for all simultaneous transmissions from $v_k, k \neq i, k \geq 0$ using directional antennas pointed in the direction of v_j the following relations hold: $\left|v_i - v_j\right| \leq r,\ \left|v_k - v_j\right| \geq (1+\beta)r, \beta > 0$, while β is a parameter designed to avoid the interference that can occur just outside of the transmission range r, otherwise transmission that violates this condition defined as a *collision*. We also assume in our model that time is slotted and one hop transmission consumes one *time slot (TS)*. A node can either transmit or receive in one TS. A one *delay time* or one *idle time* is defined as any time unit during which a message sits at a note without being transmitted. The term *schedule* is a specific indication when a node should transmit.

3 Problem Statement and Our Performance Measure

Given a network topology with N nodes, M of which have messages for BS, the goal is to find an *optimal scheduling algorithm* that schedules and routes all the messages to the BS in a minimum time (primary criterion) and also minimizes the average message-delivery time. Let $T_{end_{\min}}$ the minimum completion time for all messages to reach the destination, and T_i the time it takes for message i to reach the destination. The *delay time or idle time* Δ_i of a message m_i is a total sum of delay times that m_i incurs from t_0 until arriving at the destination. Denote by S the minimum sum of idle times for all messages. Thus the target function is: (1) $T_{end_{\min}} = \min\left(\max_{\forall m_i} T_i\right)$, (2) $S = \min\left(\sum_{i=1}^{M} \Delta_i\right)$.

3.1 Analysis – Linear (Line) Networks

Each sensor plays a role of node in the graph *G(V,E)*. The network is static and the base station (BS) is always at the end of the network. The distance between each two nodes will be denoted by d. Each node has a directional antenna with a range $d \leq r < 2d$. In fact, node only hears its left-hand neighbor in this topology. A node v_i is said to be at *i hops* from the BS. We assume a realistic situation where a node cannot transmit and receive simultaneously and the transmission can be done only in one direction (from left to right).

Following our problem definition we would like to prove that, there exists an optimal scheduling algorithm that can handle any type of a legal input x from the collection of all the possible legal inputs*X*. We denote by $d(v_i, v_j)$ the distance, measured in number of hops, between node v_i and node v_j. We define a *group of messages* as a finite set of messages, with the smallest group having only one message.

For our analysis we would like to give some helpful definitions. Let $\hat{v}_i$ be a node having a message i to transmit. *Input optimal state* denoted by $X^{opt} \in \mathrm{X}$ is a situation when the distance between any two adjacent nodes with messages to be send is at least 2 (i.e. $d(\hat{v}_i, \hat{v}_{i+1}) \geq 2$). *Optimal minimal state input* $X^{opt_m} \in X$ is a situation when the distance between any two adjacent nodes with messages to be sent is exactly 2. *Non-optimal input* $\bar{x}$ is every input that violates X^{opt} definition. Lets we define the maximum transmission rate (and also propagation of messages) of messages denoted by R as the maximal rate that the messages successfully flows toward the BS. We assume that the capacity of each node's buffer is one message (in our model, two simultaneously transmission of those messages defined as a collision). Message m_i is said to be *dependent* on message m_j in optimal solution satisfying (1) and (2) applied to $\bar{x} \in \mathrm{X}$, if m_i is not transmitted in the current time slot because we need to transmit m_j. We will denote by u_i the group of maximal length that is transmitted at the maximum rate R_i. We define that two groups u_i and u_j of maximal length are *independent* if the messages belonging to some group can be transmitted in at maximum rate without being delayed by the messages of other group. We say that groups $u_1, u_2, ..., u_k$ of maximal length are independent if messages belonging to u_i are not delayed by messages of groups $u_1, u_2, ..., u_{i-1}, 2 \leq i \leq k$. We denote this fact by $\{u_i\}_{i=1}^{k}$.

Lemma 1 Given $\overline{x} \in X$ and $\{u_i\}_{i=1}^{k}$, if the maximal transmission rate of u_i is $R_i = \frac{1}{2}$, then either $\forall \hat{v}_l, \hat{v}_{l+1} \in u_i$, $d(\hat{v}_l, \hat{v}_{l+1}) = 2$ or a group u_i has a structure of non-optimal input.

Theorem 2:

Given that, $\overline{x} \in X$, any algorithm applied on the input that starts immediate transmission of messages from the maximal lengths group and produces the output: $\{(u_1, R_1), (u_2, R_2), \ldots, (u_i, R_{i_x})\}$ always leads to optimal solution satisfying eq. (1).

Theorem 3:

Given $x \in X$ with M as the total number of messages, any algorithm that achieves the conditions of Theorem 3, also minimizes $S = \min\left(\sum_{i=1}^{M} \Delta_{m_i}\right)$.

Proof:

Since the input x can be decomposed into $\{u_i\}_{i=1}^{k} \in x$ (will be shown later), it is sufficient to show the theorem for a single u_i that contain M_i messages, $1 \le M_i \le M$.

The algorithm is applied in parallel on all u_i and each message is transmitted immediately at the maximal possible rate. Let T_{m_i} be the time it takes for a message $m_i \in u_i$ to reach the destination. We will prove that any algorithm achieving $\min\left(\max_{\forall m_i \in u_i} T_{m_i}\right)$ and satisfying condition of Theorem 2 also produces $\min\left(\sum_{i=1}^{M_i} \Delta_{m_i}\right)$. We distinguish between two cases.

Case A: When a configuration of u_i is optimal then $\sum_{i=1}^{M_i} \Delta_{m_i} = 0$, and the proof follows immediately.

Case B: When the configuration u_i is non-optimal, we proceed in the following way. First we make some useful notations. Let $\widetilde{T}_{u_i}$ be the time it takes for all messages in a group u_i to reach the destination (the completion time of u_i), and $\widetilde{t}_{m_i}$ be the time it takes for a message $m_i \in u_i$ to reach the destination without any delays. Then, the following holds: $\widetilde{T}_{u_i} = \max_{\forall m_i \in u_i} T_{m_i}$, $\Delta_{m_j} = T_{m_j} - \widetilde{t}_{m_j} \ge 0, \forall m_j \in u_i$. Since u_i is non-optimal input it follows from Lemma1that messages from u_i will be transmitted at rate ½ and also will be arrived at destination with rate ½. Therefore, if the last message m_{M_i} from u_i arrives to destination at time $T_{m_{M_i}} = \widetilde{T}_{u_i}$ then, the one before last message arrives to destination at time $\widetilde{T}_{u_i} - 2$, and so on. Thus, we have $\Delta_{m_{M_i}} = \widetilde{T}_{u_i} - \widetilde{t}_{m_{M_i}}$, $\Delta_{m_{M_i-1}} = \widetilde{T}_{u_i} - \widetilde{t}_{m_{M_i-1}} - 2$, $\Delta_{m_{M_i-2}} = \widetilde{T}_{u_i} - \widetilde{t}_{m_{M_i-2}} - 4 + \ldots$ .If we sum up delays for all messages in u_i we get the following: $\sum_{i=1}^{M_i} \Delta_{m_i} = M_i \widetilde{T}_{u_i} - \left\{(M_i - 1)\frac{M_i}{2} 2 + \sum_{i=1}^{M_i} \widetilde{t}_{m_i}\right\}$. We notice that $\sum_{i=1}^{M_i} \widetilde{t}_{m_i}$ is a constant term as well as $(M_i - 1)\frac{M_i}{2} 2$. Therefore, by minimizing the term $\widetilde{T}_{u_i}$ we minimize the total sum of delays produced by any algorithm that achieves $T_{end_{\min}}$ and satisfies conditions of Theorem 2 It remains to show that no other algorithm can lead to smaller sum of delays. We notice that, $(M_i - 1)\frac{M_i}{2} 2$=0+2+4+6+…2(M_i-1). These are the differences between arrival times of messages at BS. As we pointed out above, the maximum reception rate of BS is at most 1/2, thus it is not possible to get better (in terms of sum) sequence of differences between arrival times of messages at BS. So we improved the lower bound for the linear network and call any such algorithm that achieves (1) and (2) as an optimal scheduling algorithm. ■

Algorithm Linear Network: *For $\forall \hat{v}_i \in V$, transmit immediately if its neighbor to the right v_{i-1} has no message to transmit.*

Again, since the input x can be decomposed into $\{u_i\}_{i=1}^{k} \in x$ (will be shown later), it is sufficient to show the correctness of the algorithm for a single u_i. We will show below by induction that our algorithm achieves $\min\left(\max_{\forall m_i \in u_i} T_{m_i}\right)$. Let M_i be the number of messages in a group u_i. If $M_i = 1$, the only message is transmitted in each time slot until it reaches the destination. Assume that $\min\left(\max_{\forall m_i \in u_i} T_{m_i}\right)$ holds for $M_i = L$ messages and let $T_{end_{\min}}^{L} = \min\left(\max_{\forall m_i \in u_i} T_{m_i}\right)$. We will prove that $\min\left(\max_{\forall m_i \in u_i} T_{m_i}\right)$ holds for $M_i = L+1$ messages. If m_{L+1} is dependent on m_L then m_{L+1} will arrive at the destination at time $\left(T_{end_{\min}}^{L}\right) + 2$. If m_{L+1} is independent on the rest of messages in u_i then the minimal optimal time to reach the destination is $T_{end_{\min}}^{L+1} = d(BS_i, \hat{v}_{L+1})$ which is achieved by our algorithm. Thus, the algorithm reaches $\min\left(\max_{\forall m_i \in u_i} T_{m_i}\right)$.

The running time of the above algorithm (i.e. the number of tests whether a right neighbor contains a message to transmit) for all messages to reach the destination is $O(N^2)$, where N is a number of nodes in the network.

The strategy to determine $\{u_i\}_{i=1}^{k}$ is omitted due to lack of space.

Finally, we observe that in our algorithm no message can bypass another message that is closer to the destination.

3.2 Analysis – Two Branch Networks

Similar to the Linear Network case, we have the following topology:

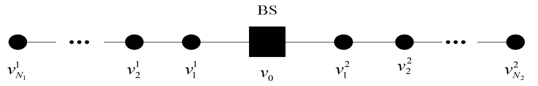

Figure 1:Two branch network

The difference from the previous case is in the location of BS: now it can be located everywhere on the line.

Denote by $V^1 = \{v^1_1, v^1_2 ..., v^1_{N_1}\}$ be a set of nodes to the left of BS (denoted by v_0) and $V^2 = \{v^2_1, v^2_2 ..., v^2_{N_2}\}$ be a set of nodes to the right of BS. The transmission can be done only in one direction: the nodes belonging to V^1 can transmit messages only from left to right, while the nodes belonging to V^2 can transmit messages from right to left. The branch containing V^1 nodes is called a *left branch (*in short, *1-branch)*, while the branch with V^2 nodes is called a *right branch* (in short, *Two-branch).* Each *j*-branch, *j*=1,2 has a legal input x^j.The goal is to transmit *M* messages to BS, $M \le N_1 + N_2$, when each node wants to transmit at most one message. In other words, 1-branch nodes transmit $M_1 \le N_1$ messages to BS and 2-branch nodes transmit $M_2 \le N_2$ messages to BS. The rest of the model as well as target functions are the same as in Linear Network case. Let us denote by T^j_i to be an arrival time of message m^j_i from *j*-branch at destination BS. The delay time or idle time Δ^j_i of a message m^j_i is a total sum of delays that m^j_i underwent until arriving to the destination BS. We are interested in optimal scheduling algorithm to minimize the following criteria: $T_{end_{\min}} = \max_{\forall m^j_i \in j-branch, j=1,2} T^j_i$ and $S = \sum_{j=1}^{2} \sum_{i=1}^{M_j} \Delta^j_i$ ·

Suppose that we apply Linear Network Algorithm from previous section to 1-branch and 2-branch independently.

Case 1: Arrival times $T^1_i, i = 1,...M_1$ are pair-wise different from arrival times $T^2_k, k = 1,...M_2$.

Case 2: There exist *i'* and *k'* such that $T^1_{i'} = T^2_{k'}$.

The case 1 is very simple, since all the messages in 1-branch are independent from the messages in 2-branch, because, the BS has only one message to serve from its adjacent node belonging to the left or the right branch.

Therefore, $T_{end_{\min}} = \max\left(T^1_{end_{\min}}, T^2_{end_{\min}}\right)$, where $T^j_{end_{\min}}, j = 1,2$ are the minimum time to reach the BS independently for 1-branch and 2-branch messages, respectively. Clearly,

$S = \sum_{j=1}^{2} \sum_{i=1}^{M_j} \Delta^j_i = \sum_{i=1}^{M_1} \Delta^j_i + \sum_{i=1}^{M_2} \Delta^j_i = S_1 + S_2$, where S_1 and S_2 are the minimum sum of idle times to reach the BS independently for 1-branch and 2-branch messages, respectively. It remains to show how to deal with the case 2. The problem is what should be the optimal policy for choosing the appropriate message to transmit? Should the message $m^1_{i'}$ scheduled before or after the message $m^2_{k'}$? From the previous section analysis it follows that all the messages from 1-branch as well as a messages from 2-branch have been partitioned into independent groups of maximal length $\{u^1_i\}_{i=1}^{s}$ and $\{u^2_i\}_{i=1}^{t}$, respectively.

<u>Lemma 4</u>

Assume that *i'* and *k'* are minimal that satisfy $T^1_{i'} = T^2_{k'}$. Assume, w.l.o.g., that algorithm decides to schedule $m^1_{i'} \in u^1_r$, $1 \le r \le s$ before $m^2_{k'} \in u^2_h, 1 \le h \le t$. The new idle time $\overline{\Delta}^2_{m_l}$ of $m^2_l \in u^2_h$ such that $d(\hat{v}_{m^2_l}, BS) \ge d(\hat{v}_{m^2_{k'}}, BS)$ is $\overline{\Delta}^2_{m_l} = \Delta^2_{m_l} + 1$·

However, as a result of delaying messages from u^2_h, a number of independent groups of maximal length from $\{u^2_i\}_{i=1}^{t}$ may become dependent and form (several) another independent group(s). In order to check this we observe that it may happen for any two adjacent groups u^2_f and u^2_{f+1} if $\delta_i = 0$ when m_{i-1} is the last message in u^2_f and m_i is the first message in u^2_{f+1}. In this case, Lemma 4 holds also for messages from u^2_{f+1}. We call the new groups as independent groups of maximal length of first order and denote them as $\{w^2_i\}_{i=1}^{z}$ (similarly, $\{w^1_i\}_{i=1}^{x}$). Then, each one of these groups can be transmitted at maximal rate of 1/2 (assuming that groups from other branch do not disturb). (Similarly, we can define independent groups of second, third order and so on).

Algorithm Two Branch Network: *For both branches compute independent groups of first order* $\{w^1_i\}_{i=1}^{x}$ *and* $\{w^2_i\}_{i=1}^{z}$. *Apply Algorithm Linear Network Algorithm from previous section to 1-branch and 2-branch independently, starting to transmit messages from both branches one by one. In the case of collision, i.e. when some messages from both branches arrive simultaneously at BS, we choose to schedule the message from the group that have the largest number of remaining messages to transmit. In the case that the both groups have the same number of remaining messages then, we choose arbitrary.*

<u>Theorem 5:</u> The scheduling produced by *Two Branch Network Algorithm* achieves the minimum completion time $T_{end_{\min}}$.

We provide a lower bound for *S*. Suppose that the total number of collisions at the BS when we apply Linear Network Algorithm to 1-branch and 2-branch independently is $\overline{m}$.

<u>Lemma 6:</u> $S \ge S_1 + S_2 + \overline{m}$.

<u>Theorem 7:</u> The scheduling produced by Two Branch Network Algorithm achieves minimum *S*

The runtime of Two Branch Network Algorithm is dominated by Linear Network Algorithm which is $O(N^2)$.

3.3 Analysis – Star Networks

This is most interesting case when we have *k* branches with joint BS. We denote by $V^1 = \{v^1_1, v^1_2 ..., v^1_{N_1}\}, V^2 = \{v^2_1, v^2_2 ..., v^2_{N_2}\}, ..., V^k = \{v^k_1, v^k_2 ..., v^k_{N_k}\}$

to be the sets of nodes on every branch, respectively and by v_0 we denote the BS. The transmission is done only in one direction: towards the BS. The branch containing V^i nodes is called *i-branch*. We will work under the assumption that no interference happens between nodes of adjacent branches, though in practice the interference might be inevitable.

However, we strongly believe that our model can serve as a good lower bound for stronger models. The goal is again to transmit *M* messages to destination BS, $M \le \sum_{i=1}^{k} N_i$, when each node wants to transmit at most one message. In other words, *i*-branch nodes transmit $M_i \le N_i$ messages to BS. Let us denote by T_i^j to be an arrival time of message m_i^j from *j*-branch at destination BS. The delay time or idle time Δ_i^j of a message m_i^j is a total sum of delays that m_i^j underwent until arriving to the destination BS. Thus, we are interested in optimal scheduling algorithm that brings to minimum the following criteria:

$T_{end} = \max_{\forall m_i^j \in j-branch,\, j=1,2,\dots,k} T_i^j$ and $S = \sum_{j=1}^{k} \sum_{i=1}^{M_j} \Delta_i^j$. Here we propose a polynomial time efficient algorithm that finds an optimal scheduling algorithm.

We start by distinguishing two cases, similar to analysis for 2-branch network. Suppose that we apply Linear Network Algorithm to any *i*-branch independently. If arrival times $T_i^1, i = 1,\dots M_1, T_i^2, i = 1,\dots M_2, \dots, T_i^k, i = 1,\dots M_k$ are pair-wise different (as the trivial case considering Two Branch Network Algorithm) then, all the messages in *i*-branch are independent from the messages in *j*-branch ($i \ne j$). Thus, the algorithm is optimal in terms of the two criteria.

When a collision occurs at the BS, we assume that $k \ge 3$ (the network consist of at least Three branches). Let us consider the following situation which is called a *static state*: suppose that we have a number $t, t \le k$ of branches $i_1, i_2, \dots, i_t$ such that (i) the closest message to BS in each such branch is at distance 1 to BS, (ii) all messages from each such branch are guaranteed to be transmitted at rate 1/2, independently, and (iii) the rest of branches do not have any message to transmit. Notice, that condition (ii) implies that each one of *t* branches contains one independent group of (at least) first order.

Let $res_j, 1 \le j \le t$ be a number of messages in i_j-branch. If $t = 2$ Two Branch Network Algorithm transmits all the messages optimally to destination BS. For $t > 2$ we have the following algorithm that solves optimally static state.

Algorithm Star Network: *If there exist* $res_q, 1 \le q \le t$ *such that,* $res_q \ge \sum_{i=1, i \ne q}^{t} res_i + 1$, *then BS first serve the q-branch and any of the rest t branches, alternately. Otherwise, let* $sum = \sum_{i=1}^{t} res_i$. *Split all the messages in two groups: A with* $\lceil sum/2 \rceil$ *messages and B with* $\lfloor sum/2 \rfloor$ *messages as following: A will contain all the messages from* i_1*-branch,* i_2*-branch,…,* i_f*-branch, such that* $\sum_{j=1}^{f} res_j \le \lceil sum/2 \rceil$, *but* $\sum_{j=1}^{f+1} res_j > \lceil sum/2 \rceil$. *In addition, A will contain (if needed) a prefix messages from* i_{f+1}*-branch that are closest to BS such that the total number of messages in A is exactly* $\lceil sum/2 \rceil$. *Group B will contain the rest of messages. BS serves any of available messages at distance 1 from BS alternately from A and B, starting from A, with the priority given to messages of* i_{f+1}*-branch, if* $\sum_{j=1}^{f} res_j < \lceil sum/2 \rceil$. *Otherwise, there is no priority in serving messages.*

Theorem 8: The scheduling produced by Star Network Algorithm achieves minimum $T_{end_{\min}}$ starting at *static state*.

Theorem 9: The scheduling produced by Star Network Algorithm achieves minimum *S* starting at *static state*.

Next, we will show that some adaptation of Star Network Algorithm provides optimal solution for minimal T_{end} and minimal *S*. Assume there is an optimal scheduling algorithm (that satisfies two given criteria) for our problem. We say that a message m_i^j is *dependent* on message $m_{i'}^{j'}$ if in the optimal scheduling strategy, m_i^j is forced to be delayed as a result of $m_{i'}^{j'}$ transmission. In order to define a *phase* we determine the dependence between pairs of messages in our input. Let us put a logical edge between each pair of messages ($m_i^j, m_{i'}^{j'}$) such that m_i^j is *dependent* on message $m_{i'}^{j'}$. A *phase* is a set of messages in the largest connected component of obtained graph with logical edges that contains the closest message to BS. We say that phase *is started*, when the first message belonging to phase is received by BS. We say that phase *is completed*, when the last message belonging to phase is received by BS. During the execution of our algorithm we may have a number of phases. Let us denote them: *1*-phase, *2*-phase, …, *k'*-phase, correspondingly. We show an optimal scheduling after applying Linear Network Algorithm to any *i*-branch independently.

Let us consider independent groups of (at least) first order $\{w_i^1\}_{i=1}^{x_1}, \{w_i^2\}_{i=1}^{x_2}, \dots, \{w_i^k\}_{i=1}^{x_k}$, similarly to the Two Branch Network Algorithm case (i.e. the messages of each group can be transmitted at rate 1/2). Each group w_i^j contains n_i^j messages. Assume that we apply Linear Network algorithm to any *i*-branch independently and a collision at the BS occurs at time $t+1$. Suppose, as before, that we have $f, f \le k$ of branches $i_1, i_2, \dots, i_f$ such that the nearest message to BS at time *t* in each such branch is at distance 1, that belong to $w_{j_1}^{i_1}, w_{j_2}^{i_2}, \dots, w_{j_f}^{i_f}$, correspondingly. Denote by $RES^t(w_{j_s}^{i_s}), 1 \le s \le f$ be a set of messages in i_s-branch in group $w_{j_s}^{i_s}$ in time *t* and by $|RES^t(w_{j_s}^{i_s})|, 1 \le s \le f$ be a number of messages in i_s-branch in group $w_{j_s}^{i_s}$ in time *t*. Let $R(t) = \cup_{s=1}^{f} RES^t(w_{j_s}^{i_s})$.

We apply the Star Network Algorithm for $R(t)$ in order to obtain optimal scheduling algorithm. Notice as time *t* grows up at each step, the set $R(t)$ undergoes changes and at each step we apply Star Network Algorithm to an updated set $R(t)$.

Observation 1

If optimal scheduling algorithm completes transmitting the last message of *i*-phase with *x* messages in time *t*, then our algorithm also completes transmitting the last message of *i*-phase in time *t*.

Suppose that the first message of *i*-phase has been transmitted by the optimal scheduling algorithm in time *t'*. In order to see the correctness of the above observation we distinguish between 2 cases: (a) *t>x+t'-1*, and (b) *t=x+t'-1*. For case (a) it means that either (i) there is a branch with a number of messages that is larger than or equal to *x/2+1* at time *t'* or (ii) starting from some critical time $t''(t'' > t')$, there is a branch with a number of messages that is larger than the total number of messages in other branches, i.e. larger than $|R(t'')|/2$ and its happens after some critical time *t''*. The inequality for case (a.i) holds during all the steps of *i*-phase. Therefore, we have to complete with the last message of *i*-phase in this branch. Since Star Network Algorithm leads to maximizing a number of messages transmitted at rate 1, the claim follows. Regarding case (a.ii) we point out that even if the critical time for Star Network Algorithm is smaller than *t''*, still the last message of this phase is transmitted from the same branch and at the same time. This is due to the fact that at each step, Star Network Algorithm maximizes the number of messages transmitted at rate 1. For case (b), it means that all messages in optimal scheduling algorithm were sent at rate 1. Since the main property of Star Network Algorithm is to maximize a number of transmitted messages at rate 1, we obtain the desired result.

Theorem 10: The scheduling algorithm explained above achieves minimum T_{end} and minimum *S*.

We notice that we can maintain on the fly the set $R(t)$ and know the exact number of messages in $RES^t(w^{i_s}_{j_s}), 1 \le s \le f$. Thus, the overall running time is $O(kN^2)$, where *k* is the number of branches and *N* is the number of nodes in the network.

4 References

[1] J. R. Argy, 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.

[2] A.Sen and M.Huston, "A new model for scheduling radio paket networks", *Wireless networks*, Vol. 3, pp. 77-82,1977.

[3] J. Hammond and H. Russell, “Properties of a Transmission Assignment Algorithm for Multiple-Hop packet Radio Networks”, *IEEE Transactions Wireless Communications*, Vol. 3(4), pp. 1048-1052, 2004.

[4] I. Rhee and J. Lee, “Distributed Scalable TDMA Scheduling Algorithm”, NCSU Technical Report 2004-4-14, Department of Computer Science, North Carolina State University.

[5] A. Chou and V. Li, “Fair Spatial TDMA Channel Access Protocols For Multihop Radio Networks”, *Proc. IEEE INFOCOM*, pp. 1064-1073, 1991.

[6] S. Ramanathan, “A Unified Framework and Algorithm for Channel Assignment in Wireless Networks”, *Wireless Networks*, Vol 5, pp. 81-94, 1999.

[7] G. Wang and N. Ansari, “Optimal Broadcast Scheduling in Packet Radio Networks Using Mean Field Annealing*”, IEEE Journal on Selected Areas in Communications*, Vol. 15(2), pp. 250-260, 1997.

[8] X. Ma and E. Lloyd, “An Incremental Algorithm for Broadcast Scheduling in Packet Radio Networks”, *Proc. IEEE MILCOM*,

[9] X. Ma and E. Lloyd, “A Distributed Protocol for Adaptive Broadcast Scheduling in Packet Radio Networks”, *Proc. ACM/IEEE DIALM for mobility*, 1998.

[10] I. Chlamtac and A. Farago, “Making Transmission Schedules Immune to Topology Changes in Multi-Hop Packet Radio Networks”*, IEEE/ACM Transactions on Networking*, Vol. 2(1), pp. 23-29, 1994.

[11] I. Chlamtac and S. Kutten, “A Spatial Reuse TDMA/FDMA for Mobile Multi-Hop Radio Networks”, *Proc. IEEE INFOCOM*, Vol. 1, pp. 389-394, 1985.

[12] F. Ali, P. Appani, J. Hammond, V. Mehta, D. Noneaker and H. Russel, “Distributed and Adaptive TDMA Algorithms for Multi-Hop Mobile Networks”, *Proc. IEEE MILCOM*, pp. 546-551, 2002.

[13] A. Ephremides and Truong, “Scheduling Broadcast in Multihop Radio Networks*”, IEEE Transactions on Communications*, Vol. 38(4), pp. 456-460, 1990.

[14] X. Ma and E. Lloyd, “Evaluation of a Distributed Broadcast Scheduling Protocol for Multihop Radio Networks”, *Proc. IEEE MILCOM*, pp. 1-7, 2001.

[15] I. Chlamtac and A. Lerner’ “A Link Allocation Protocol for Mobile Multi-Hop radio Networks”, *Proc. IEEE INFOCOM*, pp. 238-242, 1985.

[16] R. Liu and E. Lloyd, “A Distributed Protocol for Adaptive Link Scheduling in Ad-hoc Networks*”, Proc. IASTED International Conf. on Wireless and Optical Communications*, pp. 43-48, 2001.

[17] S. Ramanathan and E. Lloyd , “Scheduling Algorithms for Multihop Radio Networks”, *IEEE/ACM Transactions on Networking,* Vol. 1(2), pp. 166-177, 1993.

[18] L. Bao and J.J. Aceves, “Channel Access Scheduling in Ad-Hoc Networks with Unidirectional Links”, *Proc. ACM/IEEE DIALM for mobility*, 1998.

[19] C. Busch, M. Herlihy and R. Wattenhofer, “Hard-Potato Routing”, *Proc. STOC*, pp. 278-285 2000.

[20] Y. Mansour and B. Patt-Shamir, “Many to One Packet Routing on Grids*”, Proc. STOC*, pp. 258-267 1995.

[21] C. Busch, “O(Congestion + Dilation) Hot-Potato Routing on Leveled Networks”, Theory of Computing Systems, 37, pp. 371-396, 2004.

[22] C. Busch, M. Ismail, M. Mavronicolas and R. Wattenhofer, “Greedy O(C+D) Hot-Potato Routing on trees”, Technical report TR, 03-07, Rensselaer Computer Science Department, 2003.

[23] K. Herlery, A. Pietracaprina, and G. Pucci, “One-to-Many Routing on the Mesh”, *SPAA*, pp. 31-37, 2001.

[24] C. Busch, M. Herlihy and R. Wattenhofer, “Randomized Greedy Hot-Potato Routing”, *Proc. SODA*, pp. 458-466, 2000.

[25] S. Rajasekaran and T. Tsantilas, “Optimal Routing Algorithms for Mesh-Connected Processor Arrays”, *VLSI Algorithms and Architectures*, pp. 411-422, 1988.

[26] S. Dolev, E. Kranakis and D. Krizanc, “Baked Potatoes: Deadlock Prevention Via Scheduling”, *Proc. PODC*, pp. 210-219, 1996.

[27] A. Ben-Dor, S. Halevi and A. Schuster, “Greedy Hot-Potato Routing”, *Proc. PODC*, pp. 225-234, 1994.

[28] A. Sridhran, and B. Krishnamachari, “Max-Min Fair Collision-Free Scheduling for Wireless Sensor Networks”, *Proc. IEEE on Performance, Computing and Communications*, pp. 585-590, 2004.

[29] Franic C. M Lau And S.H Zhang’ “Fast Gossiping in Squar Meshes/Tori with Bounded-Size Packet”, *IEEE transaction on parallel and distributed systems*, Vol. 13, No. 4, April 2002.

[30] D. W. Krumme and G. Cybenko and K. N. Venkataraman, “Gossiping in Minimal Time”, Technical Report 1027, Center for Supercomputing Research and Development, August, 1990.

[31] J. Gronkvist, “Traffic Controlled Reuse TDMA in Multi-Hop Radio Networks”, *Proc. IEEE PIMRC*, pp.1203-1207, 1998.

[32] C. Florens, and R. McEliece, “Lower bounds on data collection time in sensory networks”, *IEEE Journal on Selected Areas in Communications*, 22(6), pp. 1110--1120, 2004

[33] T. S. Rappaport, "*Wireless communications*", Principles and practices, Upper Saddle River, N.J., Prentice Hall. 1996.