Improved Algorithms for Data-Gathering Time in Sensor Networks II: Ring, Tree and Grid Topologies*

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

Abstract

We address the problem of gathering information in sensor webs consisting of sensors nodes, where in a round of communication sensor nodes have messages to be sent to a distant central node (called the base station) over shortest path. 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 take into account the fact that interference can occurs at the reception of a message at the receiver sensor. In order to save redundant retransmissions and energy, we assume a known distribution of sources (each node wants to transmit at most one packet) and one common destination. We provide a number of scheduling algorithms jointly minimizing both the completion time and the average packet delivery time. We define our network model using directional antennas and consider Ring, Tree, and Grid Network (and its generality) topologies. All our algorithms run in low-polynomial time.

Key words: Scheduling algorithms, Optimization problems, Half-duplex One-port model.

1 Introduction

Recent advances in commercial IC (Integrated Circuits) fabrication technology have made it possible to integrate signal processing and sensing in one integrated circuit. These devices are popularly known as

* The preliminary version of this paper has been appeared by ICNS'07 (Intl. Conf. on Networking and Services, 2007). The work by Yoram Revah has been partially supported by the Kreitman fellowship for excellence and Israeli Ministry of Science fellowship for PhD studies. The work by Michael Segal has been partially supported by Israeli Ministry of Trade (consortium REMON for 4G communications) and INTEL.

wireless integrated network sensors (WINS) and include micro-electromechanical systems (MEMS) technology components such as sensors, actuators, RF component and CMOSS building blocks. WINS combines micro-sensor technology and low power computing and wireless networking in a compact system. Sensor nodes are dispersed over the area of interest and are capable of RF (radio frequency) communication, and contain signal processing (DSPs) engines to manage the communication protocol and data processing before transmission. The individual nodes have a limited capability, but are capable of achieving a large task through coordinated effort in a network that typically consists of hundreds to thousands of nodes.

Networks of such devices can autonomously perform various sensing tasks such as environmental (seismic, meteorological) monitoring and military surveillance, enemy tracking, target detection, distribution of timing and position information and multi-hop communication [1]. The sensor could be sensing temperature, pressure, oil leak, radiation, etc. Generally, 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. In other words, the primary application of such networks has been in disaster relief operations, military use, conferencing and environment sensing.

A typical application in web is gathering of sensed data at distant central processing system named the base station (BS) (or the root node in the network graph). This rood node is assumed to be with greater computational, storage, and transmission capabilities than the rest of the nodes in the network. The root node typically serves as an entry point to the sensor network, integrating the sensor network with wired network. In each round of this data gathering application, all the data from all nodes need to be collected and transmitted to the BS, where the end-user can access the data. In some sensor network applications, data collection may be needed only from a region and, therefore, a subset of nodes will be used. A simple approach to accomplishing this data gathering task is for each node to transmit its data directly to the BS. Since the BS is typically located far away, the energy cost to transmit to the BS from any node is quite high. Therefore, an improved approach is to use few and multi-hop transmissions as possible to the BS. In contrast, in many emerging and envisioned applications, sensor networks will be both distributed and wireless (in terms of communication and power) [2]. Distribution is necessary for improving sensing quality: when the precise location of a signal is unknown, then distributed sensors will allow sensing to take the place closer to the event of interest than by any signal sensor. Distribution also improves robustness to environmental obstacles, which is especially crucial in situations where sensing requires line-of-sight. The sensor units, thus, need to rely on finite, local energy sources and wireless communications channels. Finally, shorter range communication is generally much cheaper than longer range communication because the radio-signal power can drop off with a quadratic power of distance [3]. As a result, it is much cheaper to transmit information using multi- hopping among sensor units.

One way to reduce the amount of data that must be transmitted (and reduce energy) in radio networks is scheduling forwarded 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. Then, we aim to solve the problem for a given certain topology of radio network and a network model, initial information (messages) located at some nodes and a single designated destination. We consider the Ring and the Tree networks and give optimal scheduling solutions that achieve a minimum completion time as well as a minimum average delivery time. For the Grid network topology (and its extensions) we propose an approximation algorithm to our problem providing 1.5 approximation ratio for maximum completion time. We present a low-polynomial time solutions for our problem for above mentioned network topologies and we also provide some useful insights.

Our research can be practically implemented in those networks: for example, whenever a node has a packet to transmit, it sends a very short message (to save battery energy) called a “Schedule Request” to a central computer (BS) that serves as the only destination in the network. The requests can be sent over an upstream control channel (or multiple upstream control channels) using, for example, ALOHA or CDMA random-access schemes. The base station is assumed to have full information about the input and the network topology. It produces a schedule and periodically transmits the schedule requests (called a MAP message) to the nodes that requested to send. This is done over a separate downstream control channel. Note that the separation in channels is between the downstream and upstream control channels and the channel in which the data messages are transmitted.

In our previous work [4], we consider linear, two-branch, and star (or multi-branch) network topologies. For each topology we provided an optimal schedule for routing all the messages to the base station, jointly minimizing both the completion time and the average packet delivery time, while all our algorithms run in polynomial 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 [44] 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. 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.

Our problem was partly addressed over the past few decades. A number of works (see [5–21]) 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, and the suggested solutions are heuristic approximation algorithms. Other works (see, e.g. [22–29]) dealt with our problem considering Grid and Tree topology, but under other (weaker) network models. For example, authors in [22] used the same target function as we suggest, but the discussion is based on several variations of “hot potato” routing. In this model each node can successfully receive and transmit more than one message simultaneously. This is a completely different model from the classical radio network model, which we chose to apply in our analysis. Furthermore, papers that are concerned with “hot potato” routing offer upper and lower bounds on performance in terms of order of magnitude, while in our work, we produce exact results or tight bounds to our problems. Similarly, other papers (see [30-34]) assume the same target function to minimize the completion time and the model under which each node can successfully receive and transmit more than one message simultaneously is explored in Ring, Tree, Grid networks and general graphs. Y. Choi et al. [35] presents a protocol for routing data messages from any sensor to the base station in a sensor network two-dimensional grid topology, by using and maintaining a spanning tree with root serving as the base station completely ignoring the interferences. Sridhran and Krishnamachari [36] presented 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 [37] and Krumme et al. [38] also study the gossiping problem of communicating a unique item from each node in a graph to every other node under 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.

In their [37,38] notations, we assume a network model denoted H1 or called “The half duplex one port model”, since this model of communication makes the weakest assumptions about both hardware and software capabilities. Gronkvist [39] assumes a stochastic model for the general network topology problem and presents a number of results under this model. Some other papers (see [40-43]) transformed a network into an undirected graph *G(V,E)* with *V* as the set of nodes and *E* as the set of edges and modeled the transmission area and the interference area as balls in graph by introducing two parameters: d_T, the transmission radius and d_I the interference radius with $d_I \geq d_T$. They deal with gathering information in specific radio networks: Line, Grid, with the same target function of minimum completion time, ignoring the requirement of minimizing average delivery time and using omni-antenna. They also show [43] that in the case of general network the problem is NP hard. Finally, Florens and McEliece [44] 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 considering the idle time of the messages and produces unnecessary dependences among messages. This, consequently, causes unnecessary delays for messages. For example, it is unreasonable not to transmit a message toward the destination if it can be transmitted without any delay. They [44] also do not provide any time-complexity analysis of their algorithms.

This paper is organized as follows: First we explain in details the network and channel model with a precise definition of our problem. Next, we address the Ring Network case. After that, we consider the Tree Network problem. The optimal scheduling strategy under both target functions for Ring network and Tree Network is explained in Sections 3.3 and 3.4, respectively. Finally, we consider a Grid Network and its extensions. We propose an approximation algorithm to our problem providing 1.5 approximation ratio for maximum completion time, where the approximation bound holds for any BS location in the Grid Network. We conclude the paper with directions for further research.

2 General Network and Channel Model

A sensor or ad-hoc network is modeled as a directed graph *G(V,E)* with *N* nodes (in the case of Grid network $N \times N$ nodes), *V* is a set of nodes, each of them representing a communication device, where each node $v \in V$ is a sensor that can transmit and receive data; *E* is a set of edges connecting nodes. There is an edge between node *v* and *w* if and only if *v* can hear *w*'s transmissions when *v* points its directional transmission antenna towards *w*. The network has a special node v_0, the *Base Station* (*BS*), that serves as a destination for all messages. This node is assumed to be the root of the graph with large computational, storage, and transmission capabilities. The root node typically serves an entry point to the sensor network, integrating the sensor network with an external wired network.

- We assume that at time $t=t_0$, each node $v \in V$ has at most one message to transmit to the destination. This is referred to as a *legal input*.
- 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 that every node in the network including the BS has the same transmission power r and that a node can not transmit and receive message simultaneously. In our model we assume that the capacity of each node's buffer is one message.
- We also assume the use of directional antennas. The signal from node v to node w propagates in a straight line in the direction of node w without dispersing to other directions. We also assume that if a message arrives successfully to the receiver, the receiver can send an acknowledgment to the sender using directional antenna on a separate channel.

Based on the above, the conditions for a successful transmission are: $\forall v, w \in V$ a message from node v that is transmitted to node w , arrives successfully at node w if for all simultaneous transmissions from $\forall u \in V, u \neq v, w$ using directional antennas pointed in the direction of *v* the following relations hold: $|v-w| \leq r,\ |u-w| \geq (1+\beta)r,\ \beta > 0$ (here *v* stands for the location of node *v*). We also assume in our model that time is slotted and one hop transmission consumes one *time slot (TS)*. As we mentioned above node can either transmit or receive in one time slot. This model of channel is called in the literature S-TDMA channel model. In summary, we model our network by a rooted graph, where the root represents the BS and an edge represents an existing wireless connection (a link) between two stations. A necessary condition for connection existence between *v* and *w* is the fact that *v* and *w* are at distance between them of at most *r*. We denote by $d(v,w)$ the distance, measured in number of hops, between node *v* and node *w*.

3 Problem Statement and Our Performance Measure

In this section we carefully define our problem. We are interested in solving the problem for various network topologies: Ring, Binary Tree and Grid networks.

3.1 Problem Statement

Assuming some network topology with N nodes (or $N \times N$ nodes in the case of Grid network), M of which have messages to be send to BS (each node has to transmit at most one message), and assuming our network model with the fact that BS is receiving the requests for transmission from the nodes that have a message to send to the BS on separate, collision free channels, the purpose is to find an 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 (a secondary criterion, which is equivalent to minimization of the sum of the message idle times).

3.2 General Target Functions

We wish to find a scheduling and routing solution for every possible input set of messages to destination. We denote by $T_{end_{\min}}$ the minimum time for all messages to reach the destination, and by T_i the time it takes for message m_i to reach the destination. The *delay time or idle time* Δ_i of a message m_i is a total sum of delays that m_i incurs starting at t_0 until arriving to the destination. Denote by S the minimum sum of idle times for all messages. Thus,

$$T_{end_{\min}} = \min\left(\max_{\forall m_i} T_i \right) \qquad (1)$$

$$S = \min\left(\sum_{i=1}^{M} \Delta_i \right) \qquad (2)$$

3.3 Analysis – Ring Network

First, we investigate a *Ring Network* topology, with each sensor playing a role of node, see Figure 1.

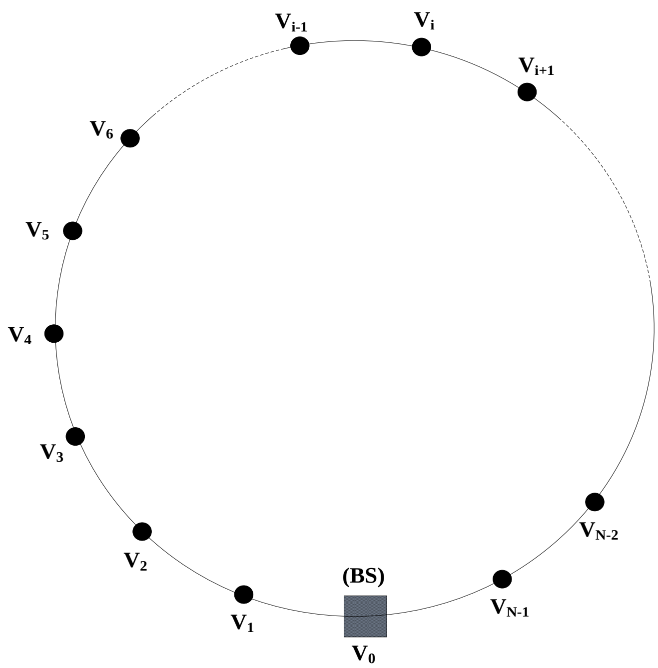

Figure 1: The Ring Network.

The Ring consist of *N* nodes including the BS and each node denoted by $v_i, i = 0,1..N-1$. The network has *M*, $M \le N-1$ messages to transmit to the BS. In this topology, it is obvious that the transmissions towards the BS can go two ways: clockwise or counterclockwise, i.e. node v_i can transmit either to node v_{i-1} or to node v_{i+1}. We also assume that the distance between any two adjacent nodes is less than or equal to *r*. Following our problem definition we would like to prove the existence of an optimal scheduling algorithm that can handle any type of a legal input (at t_0 any sensor keeps at most one message). The optimality of the algorithm is measured in terms of $T_{end_{\min}}$ and *S*. We develop the proof in stages, by proving the existence of such algorithm that schedules and routes all the messages to the BS in a minimum completion time (primary criterion) and also minimizes the average message-delivery time. We denote by $\vec{T}_i$ ($\overleftarrow{T}_i$) the minimum time it takes for message m_i to reach the destination BS using clockwise (counterclockwise) path, and by $\overrightarrow{\Delta}_i$ ($\overleftarrow{\Delta}_i$) the total sum of delays that m_i incurs starting at t_0 until arriving to the destination using clockwise (counterclockwise) direction.

We compute, for each message m_i the values $\vec{T}_i$ ($\overleftarrow{T}_i$) and $\overrightarrow{\Delta}_i$ ($\overleftarrow{\Delta}_i$) by applying *Linear Network Algorithm* [4] fixing a direction of message movement to be clockwise (counterclockwise) with respect to BS. Thus, we obtain for each message m_i the couples $\left(\vec{T}_i, \overleftarrow{T}_i\right)$ and $\left(\vec{\Delta}_i, \overleftarrow{\Delta}_i\right)$.

Algorithm Ring: *If $\vec{T}_i \neq \overleftarrow{T}_i$, send the message to the direction (clockwise or counterclockwise path) that is defined by* $\min\left(\vec{T}_i, \overleftarrow{T}_i\right)$. *If $\vec{T}_i = \overleftarrow{T}_i$ then the direction of message is defined by values $\vec{\Delta}_i, \overleftarrow{\Delta}_i$. If $\vec{\Delta}_i \neq \overleftarrow{\Delta}_i$, send the message to the direction (clockwise or counterclockwise path) that is defined* $\min(\vec{\Delta}_i, \overleftarrow{\Delta}_i)$. *If $\vec{\Delta}_i = \overleftarrow{\Delta}_i$, we can choose any direction.*

After the direction for each message is determined, we apply Two Branch Network Algorithm [4].

<u>Theorem 1</u>: The scheduling produced by *Ring Network Algorithm* schedules and routes all the messages to the BS in a minimum completion time (primary criterion) and also minimizes the average message-delivery time.

<u>Proof:</u> According our model, the capacity of each node's buffer is one message. It is obvious that two different messages can not be sent to the BS in overlapping paths in different directions since otherwise there must me a node with a buffer capacity 2. As a consequence of it we will find a *pivot node message* on the Ring. The pivot node message is the message such that all the messages that are located closer to the BS (including this message) will be transmitted to BS in the same direction and the remaining messages, (if any) will be transmitted in the other direction. In other words, in the general case there are at most two groups of the messages (as we will see latter), one part will be transmitted in clockwise direction, while the other part will be transmitted counterclockwise. Lets us denote by $\vec{T}_{pnm}$ and $\overleftarrow{T}_{pnm}$ the arrival time of the pivot node message clockwise and counterclockwise, respectively.

After we apply the *Linear Network Algorithm* independently to the same input in both directions there are two cases:

Case 1: For all the messages we have $\vec{T}_i \neq \overleftarrow{T}_i$. This means that $\vec{T}_{pnm} \neq \overleftarrow{T}_{pnm}$**.** If for all *i* either $\vec{T}_i > \overleftarrow{T}_i$ or $\vec{T}_i < \overleftarrow{T}_i$ we obtain an instance of Linear Network and our problem is solved optimally. Otherwise, we have two adjacent pivot node messages (not necessary adjacent nodes). We obtain a situation with one pivot node message with a group of messages in counterclockwise direction towards BS and the second pivot node message with a group of messages in clockwise direction towards BS. In this case, our *Ring Network* algorithm is equivalent to *Two Branch Network* algorithm [4], since we have two separate groups of messages that have to be routed to the BS over two different paths (lines) [4]. By applying *Two Branch Network Algorithm* [4], we achieve both (1) and (2) criteria. Then, $T_{end_{\min}} = \max(\overleftarrow{T}_{pnm}, \vec{T}_{pnm})$ if $\overleftarrow{T}_{pnm} \neq \vec{T}_{pnm}$. If $\overleftarrow{T}_{pnm} = \vec{T}_{pnm}$, then according *Two Branch Network Algorithm* [4], $T_{end_{\min}} = \max(\overleftarrow{T}_{tpm}, \vec{T}_{pnm})+1$, since if we decide to send two pivot node messages at the same direction we obtain $T_{end} = \overleftarrow{T}_{pnm} + 2 > \max(\overleftarrow{T}_{pnm}, \vec{T}_{pnm})+1$.

Case 2: For one message holds $\vec{T}_{pnm} = \overleftarrow{T}_{pnm}$.

In this case it is easy to see that applying *Two Branch Network* algorithm [4] gives $T_{end_{\min}} = \overleftarrow{T}_{pnm}$ *or* $\vec{T}_{pnm}$**.** However, in order to minimize the second criteria, we check the $\vec{\Delta}_i, \overleftarrow{\Delta}_i$ values and send the pivot node message in the path that guarantees minimum idle time since *Two Branch Network* algorithm in either case will postpone the scheduling of minimum group of dependent messages in the case of collision but the pivot node message will suffer $\min(\vec{\Delta}_{pnm}, \overleftarrow{\Delta}_{pnm})$ delays. ■

<u>Theorem 2:</u> Given a ring network, minimizing the sum of idle times for all messages does not lead to minimizing the total completion time.

<u>Proof</u>**:** Lets we look at more specific example (case 1, Theorem 1, for all *i*: $\vec{T}_i > \overleftarrow{T}_i$) with only two messages and the last message depends on the first message in the counterclockwise direction, thus having idle time 1.

But in the clockwise direction the message has a zero idle time despite the fact that completion time grows up.

■

The running time of this algorithm is dominated by the running time of *Two branch Network* algorithm which is $O(N^2)$.

3.4 Analysis – Binary Tree Network

The base station (BS) plays a role of the root of a binary tree graph and each sensor play a role of nodes. In the Binary Tree Network (BTN) *G(V,E,L),* root is connected to the (possibly empty) left and the right sub-trees that are also BTNs. Every such connection is a *Line Network* (see [4]), where we call the endpoints of a connection *main nodes.* In other words, main nodes in BTN are connected by line networks with sensors that serve as edges, see Figure 2. The number of the line networks in the graph denoted by L and we numerate them from the left to the right as depicted in Figure 2. We denote by v_0^{i-j} the main node that is connected in the end of line networks i, j and by v_0^{i-i} when just one line i is connected to this main node. Notice, that in this notation BS is represented by either v_0^{1-2} or v_0^{1-1}. In general, any main node can keep an arbitrary number of messages, however as we will see later it is enough to assume that the capacity of each main node's buffer is one message. However, the main node can not transmit and receive message at the same time: in the same time slot it can receive at most one message from nodes of line networks connected to it from children direction or transmit the message to the node belonging to the line network in parent direction. The main nodes, in some sense, act as relays to transfer messages towards BS. Let $V^j = \{v_1^j, v_2^j ..., v_{N_j}^j\}$ be a set of N_j nodes (sensors) in the line network numbered j (in short, j-line network) in the BTN.

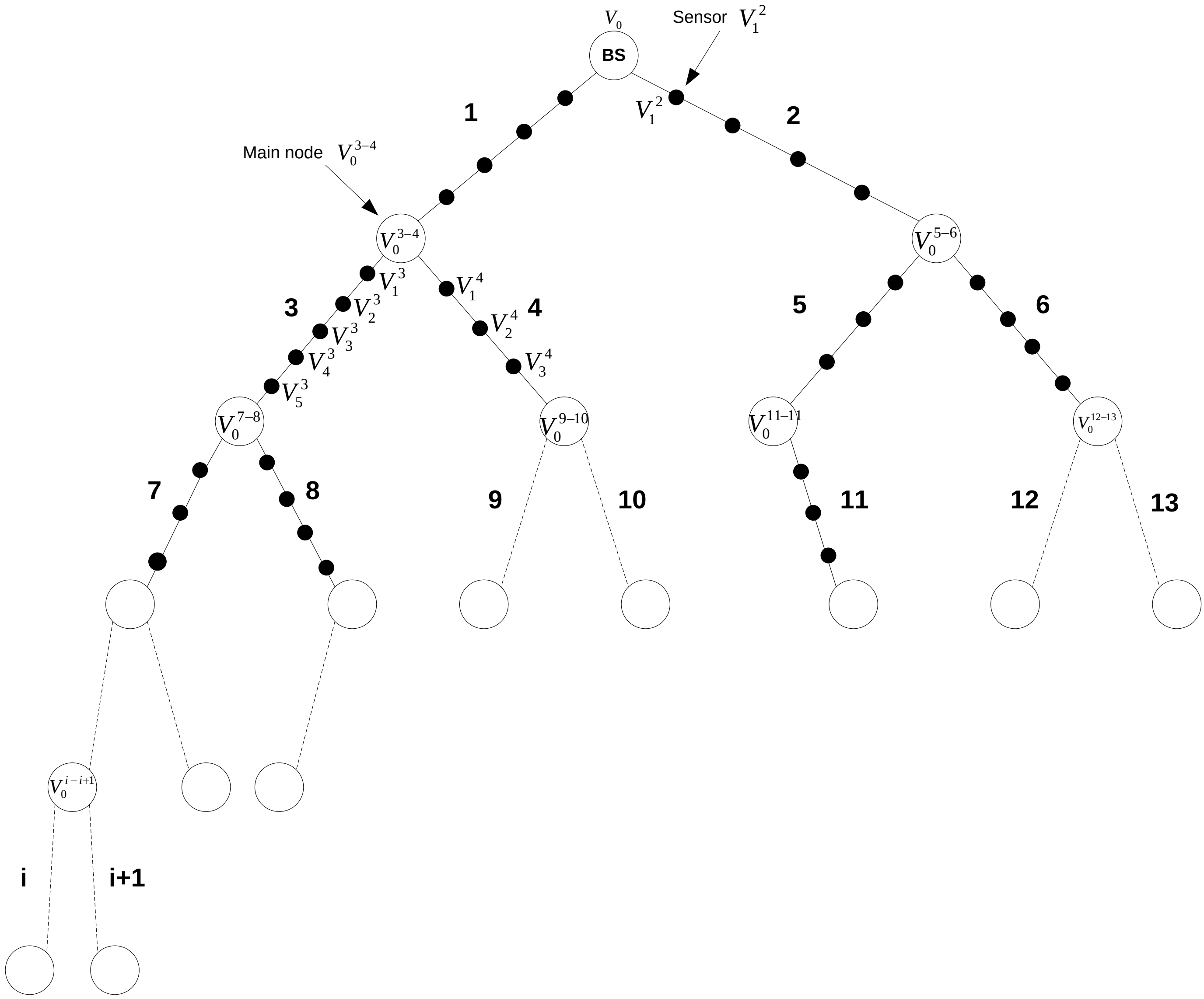

Figure 2: The BTN Network

In this topology, we define a legal input $x \in X$ as a collection of sensor nodes (exclude the main nodes and the BS) that have at most one message to transmit to the base station at $t=t_0$. The goal is to transmit M messages to the BS, $M \le \sum_{j=1}^{L} N_j$, where each line network with a legal input x_i^j , has M_j messages to transmit to the BS, while $M_j \le N_j$.

Let us denote by T_i^j to be an arrival time of message m_i^j from *j*-line network 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, using our target function, we are interested in optimal scheduling algorithm that brings to minimum the following criteria: $T_{end} = \max_{\forall m_i^j \in j-Line, j=1,2..L} \left(T_i^{\,j}\right)$ and $S = \sum_{j=1}^{L} \sum_{i=1}^{M_j} \Delta_i^j$.

Again, in order to do that, we develop the proof in stages.

Algorithm BTN : Every node in line networks behaves as normal node in line network algorithm, i.e. if some node contains a message that can be sent to its (right) neighbor (with no message), then we send it. Every main node acts as following: if it has a message, then the message is transmitted to it's parent. If the main node does not have a message to send and only one node of two possible nodes from children line network connections has a message to transmit, the main node receives it. Moreover, once main node decides a direction from which it receives messages it continue to receive them until there is no message from this direction. The main node do not serve the other line network connected to it from children direction as long as the messages from the direction that has been started to be serve by main node continue to arrive. In the case when the main node has been idle and now there are two messages from two different directions to arrive at this node, it arbitrary starts to serve one of the lines. The BS applies *Two Branch Network* algorithm [4].

Theorem 3

The *BTN Algorithm* optimizes both criteria.

Proof:

Notice that the maximum transmission rate of any main node is 1/2. This is due to the fact that according our model the main node can not transmit and receive message at the same time. The conclusion from the above observation is that in order to achieve maximum efficiency it is sufficient to supply the messages to all main nodes at rate 1/2.

We begin our explanation for one arbitrary sub-tree, say left sub-tree. We would like to "create" groups of messages of maximal length [4] that can be transmitted at the maximum rate (1/2). By using the algorithm above, according to Two Branch Network algorithm and Linear Network algorithm [4] at t_0, we have at every

j-line a number, say z, of independent groups of maximal length of first order. We denote them as $\{w_i^j\}_{i=1}^z$. In Two Branch Network algorithm [4], these groups can be transmitted to BS at maximal rate 1/2.

In addition, in the BTN, we denote by $\{\overline{w}_i^1\}_{i=1}^x$ a set of independent groups of messages of maximal length at the left sub-tree that can be transmitted to BS at maximal rate 1/2. Our algorithm produces them in the following fashion. If we have $\{w_i^1\}_{i=1}^g$ (g groups in the 1-line network) then it is easy to see that $\{\overline{w}_i^1\}_{i=1}^{g-1} = \{w_i^1\}_{i=1}^{g-1}$, and we denote by $T_{end}{}^{new}$ the time it takes the last message in w_g^1 to reach the BS while applying Linear Network Algorithm, assuming we have only the left sub tree. Afterwards, we look for the next consecutive maximal group w_1^k (e.g., this group is located in line k) closest to w_g^1. The closest means that the first message from w_1^k is at minimal distance ($d_{\min}{}^{curr}$) from BS. According to Linear Network algorithm**,** if $d_{\min}{}^{curr} - T_{end}{}^{new} \le 2$ then the groups are merged to a first order group $\overline{w}_1^1 = w_g^1 \cup w_1^k$ and simultaneously updated $T_{end}{}^{new} = T_{end}{}^{new} + 2n^{curr}$, while n^{curr} is the number of the messages in w_1^k. By this way we going down on the tree to look for the next closest group, checking again the condition $d_{\min}{}^{curr} - T_{end}{}^{new} \le 2$ and continue producing new $\overline{w}_1^1 = w_g^1 \cup w_1^k \cup .... \cup w_u^q$, where w_u^q denote the next closest group with $d_{\min}{}^{curr} - T_{end}{}^{new} \le 2$. When we finish this process, we start producing $\overline{w}_2^1$, $\overline{w}_2^1$, and so on, until we build a set $\{\overline{w}_i^1\}_{i=1}^x$. In the case that, we do not have any groups in the 1-line network, we start the process with a group of minimal $d_{\min}{}^{curr}$. The corresponding set for right sub-tree $\{\overline{w}_i^2\}_{i=1}^z$ is built in analogous fashion.

In addition, it is clear that if two any two groups of messages arrive to a main node simultaneously, the time that it takes for two groups to pass through the main node is independent on the group that the main node starts with. Finally, we obtain two logical lines with $\{\overline{w}_i^1\}_{i=1}^x$ and $\{\overline{w}_i^2\}_{i=1}^z$ inputs that it is equivalent to the case of Two Branch Network with those inputs. If we apply Two Branch Network algorithm [4] to $\{\overline{w}_i^1\}_{i=1}^x$ and $\{\overline{w}_i^2\}_{i=1}^z$ inputs, we minimize both criteria as proved in [4]. ■

Remark 1: Notice that main nodes may have messages at time t_0. The algorithm remains the same.

Remark 2: The above scheme can be generalized to deal with *k*-ary trees. The only difference is that we construct independent groups of messages of maximal length for *k* sub-trees and BS applies a strategy of *k-Star Network* algorithm.

The running time of the above scheme is dominated by the running time of Two Branch Network algorithm, which is $O(N^2)$, with N standing for number of nodes.

3.5 Analysis – Grid Network

In this section we will address the problem of a grid topology. The network has NxN nodes. In our model definition each node can do at most one operation at a given time slot, meaning that it can not transmit and receive a message at the same time; it can at most transmit one message or at most receive one message. There are two kinds of node as depicted in Figure 3: Sensor and Relay nodes. The sensor nodes are located in the sensing zone. The sensor node has the same function as in the previously discussed networks: it senses the information which is transformed to a one message to be transmitted to the BS. Relay nodes, are located outside the sensing zone on the right and the down border of the *Grid Network.*

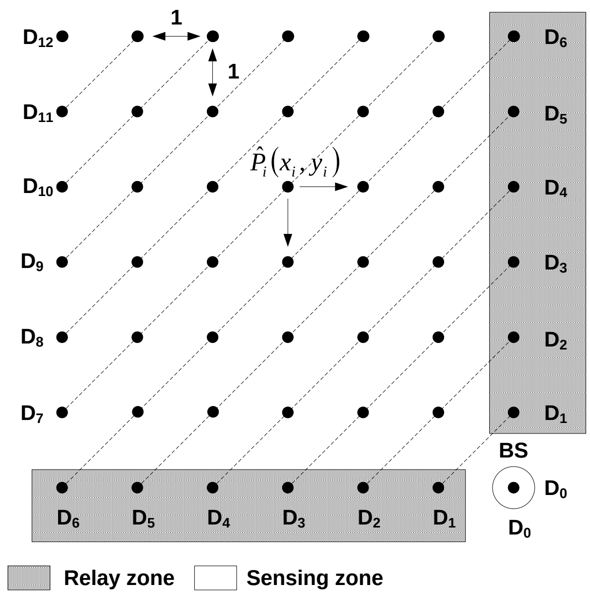

Figure 3: Grid Network

The relay node acts as a messages deliver, it just receives the messages from the sensing zone and transmits them to the BS (the relay node operates similarly to sensor node in sense that it performs only one operation per one time slot). The coordinate of node $v_{i,j}, 0 \le i, j \le N-1$ is denoted by $P_{i,j}\left(x_i, y_j\right)$ and the distance between any two adjacent nodes is the same and equals 1. For now, we will place the destination (BS) at node $P_{0,0}\left(x_0, y_0\right)$ (the relay nodes are coordinated at $P_{0,j}\left(x_0, y_j\right)$ and $P_{i,0}\left(x_i, y_0\right)$, *j=1,..,n, i=1,..,n*). In our model a message can advance to the destination only on the shortest path from the source to destination. Therefore, a message from point $P_{i,j}(x_i, y_j)$ can move to the destination only in right or down direction as marked in Figure 3. Therefore, if the length of shortest path from the source to destination (BS) is $d = x_i + y_j$ steps (hops) long, then in order to reach a destination, we need to move x_i steps in the x direction and y_i steps in the y direction. Thus, the number $\tilde{L}$ of shortest paths from source $P_{i,j}$ to destination is equal to $\tilde{L} = \binom{d_i}{x_i} = \binom{d_i}{d_i - x_i} = \binom{d_i}{y_i}$. Notice that the nodes on the same diagonal have the same distance to the destination. We also can conclude that, if there is more than one message on the same diagonal then the messages in the nodes located at this diagonal are dependent. Recall from [4] that message m_i is said to be dependent on message m_j if m_i is not transmitted in the current time slot because we need to transmit m_j. For example if we have only two messages located in the same diagonal then at least one of the messages has to wait at least one idle unit time in its way to the BS, because the other message is being transmitted. Note, that the base station can have a maximum throughput of 1, since it can receive messages in each time slot from either the relay node above it or the relay node to the left of it, alternately. Therefore, it is not possible to transfer a number of independent messages more than a number of diagonals in the grid.

In what follows we propose an approximation algorithm to our problem providing 1.5 approximation ration for maximum completion time. Before we introduce our heuristic and prove its performance and bounds we will present some definitions, observations and lemmas that will assist us in the analysis of our algorithm. Let us define by $D_{\max}$ ($D_{\min}$) the diagonal that contains at least one message at the maximum (minimum) distance from the BS and $d_{\max}$ ($d_{\min}$) be the corresponding distances, respectively. Denote by $\tilde{d}(D_i)$ the distance between any node located on diagonal D_i to the BS (*i* hops).

Observation 5

Given $x \in X$, so that any two adjacent nodes $v_{i,j}$, $v_{k,l}$ satisfy $\left|(x_i + y_j) - (x_k + y_l)\right| \geq 2$, it is possible to send all the messages to BS with $T_{end_{\min}} = d_{\max}$ and S=0.

Observation 6

Given a Grid Network of size *NxN* with *M* messages to be sent at t_0 to the BS, it always happens that $T_{end_{\min}} \geq d_{\min} + M - 1$.

It is very important to mention that for achieving this lower bound (as we will see later in the example of lemma 7), the input must enable a partition into two groups of messages with the messages belonging to one group having no affect on the movement of messages from other

We denote by $S_{[a,b]}$ a set of all messages with a distance from BS being between values *a* and *b*.

Lemma 7

Assuming a *Grid network* including *M* messages to be sent at t_0 to the BS, any scheduling algorithm fulfills the lower bound $\Omega(T_{end}) = M + d_{\min} - 1$, if it fulfils the necessary condition that:

$S_{[d_{\min},d_{\min}]} \geq 1, S_{[d_{\min},d_{\min}+1]} \geq 2, S_{[d_{\min},d_{\min}+2]} \geq 3, \ldots, S_{[d_{\min},d_{\min}+M-1]} \geq M$.

Proof:

According to $d_{\min}$ definition, it always happens that $S_{[d_{\min},d_{\min}]} \geq 1$. It is clear, that in order to reach the completion time bound of $M + d_{\min} - 1$, the BS should receive messages continuously in every time slot during the next $M - 1$ time slots. It means that the distance from the next consecutive message to be sent can increase by at most 1. The proof follows. ■

Below, we show an example that the condition described in Lemma 7 is not sufficient. Suppose we have the following input: four messages coordinated at (3, 2), (4, 2), (5, 2), (6, 2) respectively in the Grid Network. It is easy to see that this input fulfils the necessary condition of lemma 6, but according to the network model we will never succeed to achieve this bound, since in any scenario at least two close messages (at distance at most 1) will compete at the same time slot to be sent to BS. We call this situation a *collision*.

Let $\hat{P}_{i,j}\left(x_i, y_j\right)$ be a node having a message to transmit (in short, $\hat{P}_{i,j}$), and let $\hat{P}_{i,j}{}^{l}\left(x_i, y_j\right)$ be a node coordinated at node $\left(x_i, y_j\right)$ whose message is intended to be sent to destination BS towards axis $l\ (l = x, y)$, i.e. if l=x it means that the message is to be sent to BS first y_j steps towards the x axis following x_i steps towards BS. Let C_k be the arrival time of message k at BS and Δ_k denotes total idle time of message k at BS. In our heuristic, the idle time Δ_k equals $\Delta_k = C_k - d(\hat{P}_{i,j}, BS)$.

The algorithm numerates diagonals $D_i{}^{j}\ \ 1 \le j \le q,\ \ q \le 2N-1,$ that have messages to send to BS. We obtain the sorted group H=$\left\{D_i{}^{j}\right\}_{j=1}^{q}$, where $N_{D_i^j}$ is the number the messages in $D_i{}^{j}$, and $G_i{}^{j}$ denotes a collection of nodes with messages located in diagonal $D_i{}^{j}$. Assume that message k is located at i hops from BS. Then $\Delta_k = C_k - \widetilde{d}(D_i{}^{j'}) = C_k - i.$ The function max(D_i [x]) (max(D_i [y])) determines the node with a message that is located at D_i and has the maximal x coordinate (maximal y coordinate). The function $\vec{N}um(O_i{}^{j}, y)$ produces the number of nodes with the messages in the group $O_i^j \subseteq G_i^j$ that the y-coordinate of each node is less or equal to y value. Let $\overleftarrow{N}um(O_i{}^{j}, y)$= $N_{D_i^j}$ - $\vec{N}um(O_i{}^{j}, y)$.

Our algorithm treats two possible types of diagonals from H. The first type of diagonals includes diagonals that contain at least two messages. In this case, we treat this diagonal $D_i{}^{j}$ by scheduling the messages starting at the message max(D_i^j [y]) routed to the BS via $\hat{P}^{\,y}(\cdot,\cdot)$. Then schedule the message max(D_i^j [x]) to the BS via $\hat{P}^{\,x}(\cdot,\cdot)$. The algorithm continues to schedules the rest of messages from $D_i{}^{j}$ in the similar fashion, alternately, completing the entire schedule of $D_i^{\,j}$ in time slot *t*. After we complete to schedule all the messages from $D_i^{\,j}$, we check whether there is any message from $D_{i'}^{\,j+1}$ that can be sent in a consecutive time slot *t+1*. If the answer is positive, then we wait one time slot before scheduling the messages from $D_{i'}^{\,j+1}$, i.e. we start scheduling messages from $D_{i'}^{\,j+1}$ starting at time slot *t+2*. This is done in order to prevent future undesirable collisions. Otherwise (the answer is negative), algorithm continues (without any delay) the scheduling process.

After we finish scheduling messages from $D_{i'}^{\,j+1}$, we repeat the same process of inserting an artificial delay as explained above, i.e. we check whether the next message can be sent in a consecutive time slot.

The second type of diagonals includes diagonals from *H* with one message to send. Suppose we are dealing with diagonal $D_k^{\,l}$ and the message located at node $\hat{P}_{i,j}(x_i, y_j)$ from $D_k^{\,l}$ is to be scheduled at time slot *t* ($x_i + y_j = k$). In order to determine the direction of this message towards BS, the algorithm checks whether there is a message from $D_{k'}^{\,l+1}$ that can be sent in a consecutive time slot *t+1*. If the answer is negative, we can choose any direction we like. If the answer is positive, then the algorithm schedule the messages from both diagonals $D_k^{\,l}$ and $D_{k'}^{\,l+1}$ consequentially, starting with the message of $D_k^{\,l}$. If $\vec{N}um(O_{k'}^{\,l+1}, y_j) >$ $\overleftarrow{N}um(O_{k'}^{\,l+1}, y)$ then the algorithm starts routing the message located at $D_k^{\,l}$ diagonal via $\hat{P}_{i,j}^{\,y}(x_i, y_j)$. Routing of messages from $D_{k'}^{\,l+1}$ diagonal starts from the message max($D_{k'}^{\,l+1}$ [*x*]) towards *x*-axis following routing of max($D_{k'}^{\,l+1}$ [*y*]) towards *y*-axis and continues at the same fashion, alternately. If $\vec{N}um(O_{k'}^{\,l+1}, y_j) \leq$ $\overleftarrow{N}um(O_{k'}^{\,l+1}, y)$, the algorithm starts routing the message located at $D_k^{\,l}$ diagonal via $\hat{P}_{i,j}^{\,x}(x_i, y_j)$ and the messages from $D_{k'}^{\,l+1}$ diagonal are routed starting from max($D_{k'}^{\,l+1}$ [*y*]) towards *y*-axis following routing of max($D_{k'}^{\,l+1}$ [*x*]) towards *x*-axis and proceeding in alternate fashion.

After we finish scheduling messages from $D_{k'}^{l+1}$, we repeat the same process of inserting an artificial delay as explained above, i.e. we check whether the next message can be sent in a consecutive time slot.

Lemma 8:

Grid algorithm explained above satisfies criteria (1) and (2) when he input fulfills condition of Observation 5.

Proof: Since the distance between any two adjacent messages is at least 2, the algorithm does not insert any additional delay. ■

Lemma 9

The *Grid Algorithm* schedules the messages without collisions.

Proof: The algorithm schedules two types of diagonals. The first type concerns about the diagonals containing more than one message. It is easy to see that the messages from any diagonal of such kind are divided into two groups: those to be routed to the BS towards *x*-axis and those to be routed to the BS towards *y*-axis. Thus, the collisions between the messages from two groups are impossible. (For a diagonal containing only one message we even do not have groups.) In addition, no collision occurs at the BS station since the algorithm gives to the messages successive arrival time (every group arrives at BS with 1/2 rate. The second type of diagonals includes diagonals containing exactly one message. The method based on values $\vec{Num}$ ensures that no collision is possible between messages from this and following diagonal since all the messages that will be routed towards, say *y*-axis, have their nodes *y* coordinate greater (smaller or equal) than the messages that will routed towards *x*-axis. As a result, those two groups of messages are routed through disjoint paths to the BS. It remains to show that no collisions are possible between the messages of different diagonals. This follows immediately due to the fact that we insert an additional delay time between transmissions at consecutive time slots from adjacent diagonals according to algorithm. ■

Theorem 10

Given $x \in X$ with *n* messages when $T^{opt}{}_{end_{\min}}(x)$, $T_{end}^{Grid}(x)$ stand for the completion time as functions of input *x* for an optimal algorithm and *Algorithm Grid*, respectively we obtain:

$$\max_{\forall x \in X} \frac{T_{end}^{Grid}(x)}{T^{opt}{}_{end_{\min}}(x)} = 1.5$$

Proof:

According to lemma 7, $T^{opt}{}_{end_{\min}} \geq n$. Notice that our algorithm inserts additional delay time slots per at least 2 messages that are scheduled, and therefore, the total number of delays is at most *n/2*. Thus,

$T_{end}^{Grid} \leq T^{opt}{}_{end_{\min}} + n/2$. We obtain that $\frac{T_{end}^{Grid}}{T^{opt}{}_{end_{\min}}} \leq \frac{T^{opt}{}_{end_{\min}} + n/2}{T^{opt}{}_{end_{\min}}} \leq 1 + \frac{n/2}{T^{opt}{}_{end_{\min}}} \leq 1 + \frac{n/2}{n} = 1.5$. ■

Proposition 1

The second constraint condition is unbounded in Algorithm Grid

Proof:

We construct an example when we have to schedule the special input with all diagonals in the grid containing one message to send. The messages are located in the following manner: each message from odd numbered diagonal D_i is located at node $(i-1,1)$, $3 \leq i \leq 1 + 2(N-1)$, and each message from even numbered diagonal D_j is located at node $(1, j-1)$, $4 \leq j \leq 2 + 2(N-1)$. It is clear that optimal schedule that is applied to this input leads to $T_{end_{\min}} = d_{\max}$ and *S*=0. However, our proposed algorithm adds a delay after every pair of messages. The proof follows. ■

The running time of the proposed algorithm is affected by efficient maintenance of maximal and minimal coordinates in nodes with messages at each diagonal and computing values $\vec{N}um(O_i^{\,j}, y)$. All this can be done in total $O(M \log M)$ time by using minimal/maximal order heaps and balanced binary search trees, where *M* stands for the total number of messages in the entire grid.

Remark 3: The algorithm above can be generalized to work with graphs that represent connected (partial) paving of plane by grids of the same size, see for example Figure 4. Moreover, the base station can be located anywhere in the graph. The basic idea is to divide (logically) the graph into 4 quarters and determine the quarter with a closest message to BS. Then we continue to send messages to BS until we have a time slot with no message. In that case, we look for the next closest message to BS that has not been sent yet. In order to avoid collision between the last sent message and the new one we send them in a different directions (we always can do it).

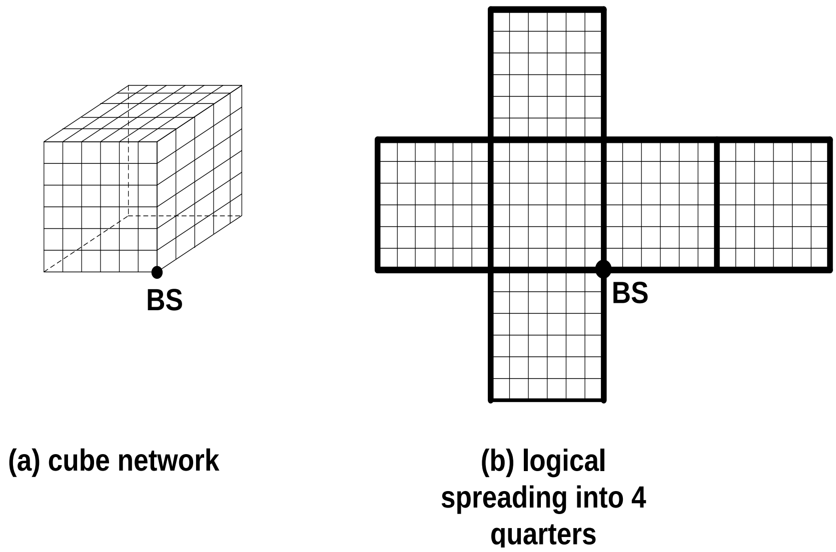

Figure 4: Cube and its planar representation

4 Conclusions

This paper continues to investigate the problem that we began at [4]: how to minimize the time completion and the sum of delays needed for gathering information at base station, from the sensor network nodes. We assume half duplex one port model equipped with directional antennas. Polynomial time algorithms for Ring Network, Tree Network and Grid Network topologies have been designed and analyzed. We provide optimal solutions for Ring and Tree networks and for Grid Network, we present 1.5 approximate solution. A possible future research includes examination of more general topology models such as general graph problem, as well as tight analysis of other target functions, e.g. including the cost of transmission depending on battery energy.

5 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] D. Estrin, L.Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks", *In Proceedings of ICASSP 2001*, Salt Lake City, Utah, May 2001

[3] U. Centintemel and A. Filinders and Y. Sun, "Power-Efficent Data Dissemination in Wireless Seensor Networks", *MobiDE' 03,* September 19, 2003, San Diego, California, USA.

[4] Y. Revah and M. Segal, "Improved Algorithm for Data-Gathering Time in Sensor Networks", accepted to Elsevier *Computer Commuications,* 2008.

[5] A. Sen and M. Huson, "A new model for scheduling radio packet networks", *Wireless Networks*, Vol. 3, pp. 77-82, 1997.

[6] 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.

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

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

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

[10] 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.

[11] X. Ma and E. Lloyd, "An Incremental Algorithm for Broadcast Scheduling in Packet Radio Networks", *Proc. IEEE MILCOM*, 1998.

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

[13] 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.

[14] 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.

[15] 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.

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

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

[18] I. Chlamtac and A. Lerner' "A Link Allocation Protocol for Mobile Multi-Hop radio Networks", *Proc. IEEE INFOCOM*, pp. 238-242, 1985.

[19] 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.

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

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

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

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

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

[25] 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.

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

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

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

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

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

[31] Y.Mansor,and B. P. Shamir, "Greedy Packet Scheduling on Shortest Paths", *Proceeding of 10th Annual ACM Symposium on Principles of distributed computing*, pp. 169-175, 1991.

[32] R. Bansal, "Online Algorithm for Routing and Scheduling on Ring Networks", *Tech Report*, Denison University, 2002.

[33] J. T. Havill," Online Packet Routing on Linear Arrays and Rings", *Proc. 28th ICALP*, July 8-12, pp. 773-784, 2001.

[34] S. Alstrup and J. Holm,"Direct Routing on Trees"' *Proc. SODA*, pp342-349, 1999.

[35] Y. Choi, M. G. Gouda, H. Zhang and A. Arora," Routing on a Logical Grid in Sensor Networks", UTCS Technical Report TR-04-49.

[36] 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.

[37] 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.

[38] 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.

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

[40] J. C. Bermond, and R. Correa, and Minli, "Gathering Algorithms on Paths Under Interference Constrains", *Proc. CIAC 2006*, LNCS, pp. 115-126, 2006.

[41] J. C. Bermond and J. Peters, "Efficient Gathering in Radio Grids with Interference" *In. Proc. ALGOTEL*, 2005.

[42] J. C. Bermond, J. Galtier, R. Klasing , N. Morales and S. Perennes," Gathering in Specific radio Networks", To appear in *ALGOTEL*, 2006.

[43] J. C. Bermond, J. Galtier, R. Klasing , N. Morales and S. Perennes," Hardness and Approximation of Gathering in Static Radio Networks", *Proc. PerCOM*, 2006.

[44] 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.