

A Tree-Based Distributed Permutation Routing Protocol in Multi-Hop Wireless Sensors Network

Alain Bertrand Bomgni, Elie Tagne Fute, Miguel Landry Foko Sindjoung, Clémentin Tayou Djamegni

Department of Mathematics and Computer Science, University of Dschang, Dschang, Cameroon Email: alain.bomgni@yahoo.fr, eliefute@yahoo.fr, miguelfoko@yahoo.fr, dtayou@gmail.com

Received 6 November 2015; accepted 20 June 2016; published 23 June 2016

Copyright © 2016 by authors and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY). http://creativecommons.org/licenses/by/4.0/

© Open Access

Abstract

A Wireless Sensors Network (WSN) is an ad-hoc network populated by small hand-held commodity devices, running on batteries called stations or sensors. Often used in hostiles and sometimes unreachable environments, stations are subject to energetic constraints which can significantly decrease the network life time. Permutation routing problem is mainly found in the literature of WSN. This problem occurs when some stations have items that belong either or not to them. The goal is to send each item to its receiver. To solve this problem, several works are presented in the literature. In this paper, we present a new permutation routing protocol for multi-hop wireless sensors network that, compared to recent work in the field is more efficient in terms of conservation of sensors' energy, which results in a longer life time of the network. Also, contrary to some other routing protocols which assume that the memory of the sensors is infinite, we show that the memory size of the sensors is limited, which in our opinion is more realistic.

Keywords

Clique, Cyclic Reception, Hierarchical Clustering, Permutation Routing Problem, Wireless Sensors Network

1. Introduction

WSN is an ad-hoc network, which is made up of small devices deployed in an area called capturing field in order to study one or more phenomena [1], [2]. Commonly monitored parameters are temperature, humidity, pressure, wind direction and speed, illumination intensity, vibration intensity, sound intensity, power-line voltage, and

How to cite this paper: Bomgni, A.B., Fute, E.T., Sindjoung, M.L.F. and Djamegni, C.T. (2016) A Tree-Based Distributed Permutation Routing Protocol in Multi-Hop Wireless Sensors Network. *Wireless Sensor Network*, **8**, 93-105. <u>http://dx.doi.org/10.4236/wsn.2016.86010</u> chemical [3]. Generally, the field of capture is an area where access is almost impossible for humans [4]. Therefore, to ensure a maximum performance of sensors, it is necessary to develop an efficient routing protocol that reduces the power consumption of the later. This is particularly important in the measure that energy spend by a sensor to send or to receive an item can be used to process a thousands of operations [5]. One of the communication techniques in the WSNs is the multi-hop communication. It ensures the sending of captured data to a station using intermediate stations. The main goal is to partition a given WSN into disjoint clique in which, a Cluster Head (CH) is elected for each clique. After that, we present a new Hierarchical permutation routing protocol which is not only energy efficient, but also reduces the work of CHs. To achieve this, we propose to use the technique of channel reservation presented in [6], the clustering technique presented in [4], and other techniques presented in [7]-[10].

1.1. State of the Art

The fundamental goal of a sensor network is to produce, over an extended period of time, globally meaningful information from raw local data obtained by individual sensor nodes. Importantly, this goal must be achieved in the context of prolonging as much as possible the useful lifetime of the network and ensuring that the network remains highly available and continues to provide accurate information in the face of security attacks and hardware failure [11]. WSNs experienced a great expansion these latest years. In fact, many works had been done in this domain; especially in the data routing and more precisely in permutation routing in a multi-hop environment. *Bomgni et al.* in [8] have introduced a deterministic routing protocol for permutation routing in dense multi-hop

sensor networks, which is realized in $\frac{n}{p} |HUB_{\max}|(k+1) + O(|HUB_{\max}|) + k^2 + 2k$ broadcast rounds in the worst

case. Lakhlef et al. proposed in [7] an improvement of the previous protocol that runs in $3n + 6\log_2(k)$ broadcast rounds. Where *n* is the number of items stored in the network, *p* is the number of sensors, $|HUB_{max}|$ is the number of sensors in the clique of maximum size and *k* is the number of cliques after the first clustering. However, the protocols presented in [8] and [9] assume that all network stations are awake during the entire execution of the protocols.

Heinzelman et al. [12] introduced a hierarchical clustering algorithm for sensor networks, called Low Energy Adaptative Clustering Hierarchy (LEACH). LEACH is a cluster-based protocol, which includes distributed cluster formation and rotation of the CH's role while transmitting data. An enhancement over LEACH protocol called PEGASIS (Power-Efficient Gathering in Sensor Information Systems) was proposed by *Lindsey et al.* in [13]. PEGASIS is a near optimal chain-based protocol in which, each node communicates only with a close neighbor and takes turns transmitting to the base station, thus, reducing the amount of energy spent per round. The protocols presented in [12] and [13] assume that the memory capacity of the sensor nodes is not limited.

1.2. Our Contribution

We consider a WSN of p stations which contains n items, each station has initially $\frac{n}{p}$ items. We propose an

energy efficient hierarchical permutation routing protocol which performs in 4 stages. The first stage is devoted to the clustering of the network into disjoint cliques where a CH is elected [4]. The second stage is the transformation of cliques obtained after the previous stage in a hierarchy of cliques using [10]. The third stage is allocated to the routing of external items toward their directed clique. Finally, in each clique, we route the internal items to their

destination. We show that according to certain parameters, each station has a memory complexity of $O\left(3\frac{n}{p}\right)$.

The remainder of this paper is organized as follow: in Section 2, we present the new hierarchical permutation routing protocol for WSNs. Section 3 deals with some experimental results. A conclusion with open problems ends the paper (Section 4).

2. New Hierarchical Permutation Routing Protocol for Wireless Sensors Network

2.1. Prerequisites

In order to properly run this protocol, it is important to define certain bases. At first, consider that the network

has p stations, each of them stores $\frac{n}{p}$ items and has a unique *ID* between 1 and p. Again, we consider that k channels of communication are available to allow the sensors to communicate. Hence, we write WSN(p,k) and we read wireless sensor network with p stations and k channels. We suppose that n items circulate in the network. Each of the p stations has $\frac{n}{p}$ items. Each item is a pair (a(v), v) $(1 \le v \le p)$. Where a(v) is the data to be send to sensor v. Item held by a station may or may not have itself as destination station. Figure 1 illustrates an example of permutation routing in a WSN of 9 stations.

2.2. First Stage: Clustering Procedure in Cliques

Clustering has become a prominent approach to reduce energy consumption in WSNs [14]. In clustering, the network is arranged in clusters of nodes, where each cluster (clique) consists of member sensors, gateways, and a cluster head. The main objective in this first stage is to partition the set of sensors of the network in cliques. Note that clustering can be done in two ways: first, the *Node-Centric* method consists of choosing the cluster head before choosing clique member's, and the second method, *Cluster-Centric*, does the reverse of the first method [8] [15]-[20]. We use the second approach to partition the network in our protocol. Indeed, we will use the protocol presented by *Sun et al.* in [4] to partition our network. At the end of this, we obtain a number of disjoint cliques within which stations are interconnected to each other, *i.e.* communication within each clique is one-hop, and each station knows the other's identity. After clustering, the members of different cliques are responsible for electing the CH. Thiselectionisdone as follows:

- 1. Each node broadcasts its residual energy with its member's clique. The sensor with the higher residual energy is then elected as CH.
- 2. If each of them has the same residual energy, the CH is the one that has the higher ID.
- 3. After sending items of a particular clique, the station whose identifier is directly below that of the current CH becomes the new CH. If the current CH has the smallest ID, the station with the greater ID will be the new CH. This is doing so on, until all cliques received their destined data.

Figure 2 shows in (a) a network of 10 not-partitioned sensors and (b) the same partitioned network into cliques (3 cliques) using the protocol presented by Sun et al. At the end of this step, all the external items of each clique are known.

2.3. Second Stage: Hierarchical Clustering

At the end of the previous stage, we obtained a set of k cliques (so k CHs). Our goal now is to constitute a hierarchical structure to ensure efficient data routing. For this, we use the protocol presented by *Banerjee et al.* in [10]. Indeed, the realization of that protocol is made as follows: starting from an initial set of sensors, a multi level covering tree cluster is constructed reassuring that each clique has a number of nodes between x and 2x, where x is an integer value parameter. To apply this, we use the k CHs obtained after the first stage. We set

 $x = \frac{k}{2}$ and we obtain a single cluster [10]. It should be noted that the obtained *superclusterhead* after the hie-

rarchical clustering knows the identity of all members of the cluster (which actually are the others CHs obtained after the first stage). Figure 3 illustrates a hierarchical clustering (1, 2, 3, 4, 5) with a parameter x = 3 given, using the protocol of *Banerjee et al.* [10].

After this stage, the cliques are in a well-defined hierarchy where it remains only to establish the order to transfer data to external cliques.



Figure 1. Permutation routing in a WSN with p = 9.





Figure 3. An example hybrid clustering: the first stage shows an example of clustering in cliques. The second stage shows the hierarchical clustering of G'.

2.4. Third Stage: Routing the External Items for Each Clique

As in the protocols presented in [8] and [9], the communication among clusters is made using gateways. In other words, when a station within a clique has the item destined to a particular station in another clique, it sends the item from gateway to gateway until it reaches the gateway of the destination clique. At this time, the latest gateway is charged to send the item at the final destination. **Figure 4** illustrates this principle. Moreover, a communication channel is assigned to each clique.

To make it simple, we consider that if x is the number of available channels and k the number of cliques obtained from stage 2, then x > k. Let's remember that at the end of stage 2, we get a tree that all nodes are CHs and leaves are sensors where one of the CHs is the CH of the tree (*superclusterhead*).



2.4.1. General Scheduling of Cliques

Our goal in this section is to present the order in which the different cliques of network receive their items. To realize this, we use a similar method like the one presented by *Bomgni et al.* in [8]. Since the *superclusterhead* knows the identity of all CHs, it broadcasts the ordered list of ID of the different CH in the tree. After receiving, each CH identifies its position in the list and informs the members of its clique. The notification of members in a clique is done in parallel within the cliques and requires 1 slot. Importantly, no station is awake for more than 1 slot. Hence, this phase requires a total of $F = 3\log_2(k) + 1$ slots and meanwhile all stations involved remain awake for up to f = 3 slots.

2.4.2. Identifying and Scheduling Cliques That Have Items in Direction of the Clique i

Phase 1: sending the list of members of the clique i to superclusterhead

The communication within cliques constructed in the first stage is one-hop, *i.e.* all stations are aware of other stations in the clique including the cluster head. The CH sends the list of members of its clique to *superclusterhead*. Therefore, this phase requires the contribution of different CHs and gateways concerned, *i.e.*, all other stations are asleep during this phase. CH of original clique and the *superclusterhead* remain awake at most 1 slot. However, the other CHs and relevant gate ways remain awake during 2 slots. One slot to receive the list and the other to retransmit. Thus, each node contributing in this phase remains awake for up to 2 slots. Communication between two cliques requires a maximum of 3 slots. One to move from one station in the clique to the gateway, another one from the gateway to another gateway and the last to move from the gateway to the corresponding station in the clique destined. The time used by the CH to send the list of its neighbors to the farthest clique is equal to $3\log_2(k)$ slots in the worst case. Due to the fact that all the cliques must receive their items and that the super clique does not transmit, this step is performed k-1 times for all the other cliques. It results to a complexity of $C_1 = 3(k-1)\log_2(k)$ slots to complete this phase. With no station being awake for more than $c_1 = 2(k-1)$ slots.

Phase 2: Broadcasting the members' list of clique i to the other cliques

Using the communication channels allocated to its son's clique in the tree, the *superclusterhead* broadcasts the list of members of the clique i. After receiving this list, the CHs record and retransmit it to its sons CHs using their assigned channels. The process is repeated until the leaves receive the list. There is $\log_2(k)$ cliques in the path from the super clique to the deepest clique. The operation described here is realized in parallel on all the branches of the tree and requires $3\log_2(k)$ slots for completion. During this phase, each cluster head or gateway involved needs to be awake for at most 2 slots, except the *superclusterhead* which only awakes to send the list and the CHs

of leaves that only wake up to receive the list in 1 slot. Then, the members' transmission of all the k cliques in the network requires $3k \log_2(k)$ time slots. Finally, this phase takes $C_2 = 3k \log_2(k)$ time slots, with no station being awake for more than $c_2 = 2k$ slots.

Phase 3: Identifying the items in destination to clique i among different cliques

The goal in this phase is to determine the number of items that different cliques have in destination to clique i. Remember that in the previous phase, the stations of different cliques have the list of stations of the clique i. Since each item to convey is the pair (a(v), v), stations know whether the items in possession are destined to clique i or not. This is done using the reservation protocol presented by *Nakano et al.* [6]. After the request of CH, the station with the lowest number wakes up at the first slot to broadcast n_1 in the reserved channel of the clique, which represent the number of item that it has in the destination of clique i. Then, all the other stations are asleep.

In the next slot, the second station with the lowest ID in the clique awakes to receive n_2 and then compute the sum $n_1 + n_2$ and broadcasts the result in the reserved channel of the clique; and during this time, it is the only station awake. The process continues up to the station with the greatest ID. Let $|HUB_{max}|$ being the number of stations in the clique with maximum size. The previous process continues up to the slot $|HUB_{max}| - 1$, where the station with the greatest number must awake to receive $n_1 + n_2 + \dots + n_{|HUB_{max}|-1}$. It therefore computes the sum $n_1 + n_2 + \dots + n_{|HUB_{max}|-1} + n_{|HUB_{max}|}$ and broadcast the result to all the other stations of the clique. Then, with the last diffusion, this operation takes a total of $|HUB_{max}|$ slots and no station is awake for more than 3 slots. One slot for the first reception of the sum, another for transmission, and the last one slot to receive the total number of items that the clique has toward the clique i. After this step, each station knows when it will wake up to forward its items. Especially, the station l must awake to the time slot $n_1 + n_2 + \dots + n_{l-1} + 1$. This operation is executed k - 1 times (the clique i is excluded from the process). Thus, this phase is realized in $C_3 = (k-1)(|HUB_{max}|)$ times slot, while

each station is awake for at most $c_3 = 3(k-1)$ slots.

Phase 4: Scheduling the cliques to transfert the items to the clique i

Previously, all stations of different cliques know the number of items destined to the clique i. The goal in this phase is to order the cliques hierarchically so that in different cliques, stations wake up at the right time to receive items from the lowest level and directed to the clique i, after which they broadcast these items to the next level and then go into sleep mode. In the first slot, CHs in leave's cliques wake up and send in the reserved channel of their clique, the number of items that it has in destination to clique i (that is, in the tree obtained in 2.3, each parent node listening over its child channel). After 3 slots, the CHs of the parents' cliques wake up and receive the numbers sent by their sons, and each of them computes the sum of these numbers. Meanwhile, all other stations not involved in the operation are asleep. That operation unfolds continuously, until the *superclusterhead* receives the number of items that all cliques except the clique i has in destination to clique i. Since different channels are used for the ascent information in the tree, it comes to *superclusterhead* in $3\log_2(k)$ slots for the most remote cliques. During this, each station involved remains awake for at most 2 slots. Overall, this phase takes $C_4 = 3(k-1)\log_2(k)$ times slots, and each station involved remains awake for $c_4 = 2(k-1)$ slots at most.

Lemma 1

Globally, the scheduling and identifying of cliques that have items directed to the clique i require $C = 3(3k-2)\log_2(k) + (k-1)(|HUB_{max}|)$ times slots to run. Each station involved in this phase remains awake for c = 9k-7 slots at most.

Proof: The proof of this result is trivial. In fact, the phase 1 of this step requires $C_1 = 3(k-1)\log_2(k)$ slots and $c_1 = 2(k-1)$ awaking slots for stations involved. Phase 2 requires $C_2 = 3k \log_2(k)$ times slots and $c_2 = 2k$ awaking slots for stations involved. Phase 3 requires $C_3 = (k-1)(|HUB_{max}|)$ slots and $c_3 = 3(k-1)$ awaking slot for stations involved. Finally, phase 4 requires $C_4 = 3(k-1)\log_2(k)$ slots to complete and $c_4 = 2(k-1)$ awaking slots for the stations involved. Whereby, computing $C = C_1 + C_2 + C_3 + C_4$ and $c = c_1 + c_2 + c_3 + c_4$, we have the result.

2.4.3. Sending External Items Identified in the Tree to the Clique i Using the Cyclic Reception Technique

All the stations in different cliques, know the exact number of item that must be broadcast from their clique to

clique i at the end of the previous step. The goal now, is firstly to send all items destined to the clique i in the super clique. Then, secondly to send these items to clique i. Let us clarify these phases.

Phase 1: Broadcast of items destined to clique i to the super clique

Our job in this phase is to forward the items destined to the clique i to the super clique. Remember that at the end of phase 3 (2.4.2), each station knows the exact time that it will wakes up to forward items of clique i. Particularly, the k^{th} station of the clique must wakes up to the time slot $t + n_1 + n_2 + \dots + n_{k-1} + 1$ to begin its transmission. Where t is the number of slots that the protocol has already consumed and n_i ($1 \le i \le k-1$) are the numbers of items held by k-1 number of stations less than k in the clique.

1) Transfer clique i items to the clique of upper level in the tree

This step runs in parallel in all the sons' cliques with same level in the tree that has items which are destined to clique i. Moreover, station with small ID awakes to the first slot and forwards items in possession for the clique i, at the top level in accordance with the channel allocated earlier. After this, the second station with the lowest ID performs its forwarding of the items destined to the clique i. The process continues until all stations in the current clique send the clique's i items to the parent clique in the tree.

2) Cyclic reception of clique's i item from low level cliques in the tree

After phase 4 of Section 2.4.2, all cliques know when they will start receiving items which are destined to clique i and arising from their low level cliques. In addition, all stations know exactly how many items they will receive from cliques of lower-level. Especially, let k be the number of the current clique, n_{k-1} the number of items coming from its son's cliques, and p_k the number of stations of the clique k. If $n_{k-1} \mod (p_k) = 0$ then, each of the stations of the clique k will receive $n_{k-1} div(p_k)$ items. Otherwise $(n_{k-1} div(p_k) \neq 0)$, the $n_{k-1} \mod (p_k)$ first clique stations will receive $n_{k-1} div(p_k) + 1$ items and the remaining stations of the clique $n_{k-1} div(p_k)$ items.

The reception of these items is as follows: in the suitable time slot, the station with lowest ID of the clique wakes up and receives the first item that arrives. Then returned in sleep mode. The second lowest ID station wakes up, receives the second item and then also returns to sleep mode. The process unfolds in the same way until the station with largest ID in the clique receives its item continuously until all nodes of lower-level cliques transmit its items.

The process of broadcasting and receiving continue so on, until the stations of the super clique receive all items destinated to the clique i. Note that for an item moves from a clique's leave to the super clique, $3\log_2(k)$ slots time are necessary in the worst case. On the other hand, in the worst case, all items of the clique i are held by clique's leaves of the tree. For this purpose, k-1 cliques contribute because the *superclusterhead* is not con-

cerned. Thus, this phase requires $D_1 = 3(k-1)\frac{n}{p}|HUB_{\max}|\log_2(k)|$ slots while each station involved stays awake

for $d_1 = 2(k-1)\frac{n}{p}|HUB_{\max}|$ slots at most.

Phase 2: broadcast of items destined to clique i from the super clique

After phase 1, all items destined to the clique i are in the stations of the super clique. The goal in this phase is to show how these items will be sending to clique i. To solve this problem, we first determine the path from the super clique to clique i. Thereafter, we inform the cliques that are on this path, and finally we broadcast items properly. These processes are illustrated by the following sections.

1) Determination of the path from the super clique to clique i

The leitmotiv here, is to determine the path from the super clique to the clique i. Recall that after the second step (Section 2.3), the *superclusterhead* knows the identity of all clusters head in the network. This simply means that to broadcast clique i items, superclusterhead knows cliques that are on the path between it and clique i, but CHs located on this path does not have this information. Therefore, the *superclusterhead* should inform the CHs about the way to move toward clique i. The CHs in turn inform the other member of their clique about that path.

2) Inform the CHs about the path toward clique i

Superclusterhead knows the path to broadcast the items to clique i. Therefore, it will inform the CHs of low level in the tree about the time they will wake up to collect the items of clique i. To start, all the network's stations are sleeping. The superclusterhead wakes up and sends to its first level sons in the tree a message containing two information: the number N_i of items destined to clique i and the path to the clique i. At receiving, the CHs record the message and forward itto their own first level sons in the tree. This is done recursively until the CHs leaves of the tree. This process requires $3\log_2(k)$ time slots. This operation takes place k-1 times, then for all cliques,

it requires $D_2 = 3(k-1)\log_2(k)$ slots. The CHs and gateways involved remain awake for at most $d_2 = 2(k-1)$.

3) Broadcast the items to clique i

The path to the clique i was determined before. We broadcast its items using the sub-step below. Remember that all the stations on the path to the clique i know at which time their clique must receive the items from their upper clique in the hierarchy.

a. Broadcast the items from super clique to clique i

To start, the station with the smallest ID in the super clique and the station which has the least ID in the son's clique of level 1 get awake (Evidently with gateways involved). The one in the super clique starts sending the items to the one in the son's clique. First, the one in the super clique send all its items destined to clique i. Second, the next station with the lowest ID does the same and so on until all the stations in the super clique have send the items destined to clique i. The broadcast is done from the super clique to its son's clique of first level belonging to the path to clique i. And the process is repeated from that son until the clique i.

b. Cyclic reception of items by cliques located on the path to the clique i

The reception of items in this sub-phase is done in the following way in all the cliques that are on the path linking super clique to clique i (including clique i): station with the smaller ID gets awake, receives the first item sent and returns to sleep mode. Then, the second station with the lowest ID wakes up, gets the next item sent, and also returns to sleep mode. The work is repeated until we get to the station with the highest ID and then the process is restarted. This takes place as long as items to receive from the higher-level cliques exist.

Firstly, the broadcast of items from the super clique to the farthest clique requires $3\log_2(k)$ slots. Secondly, in the worst case, all items in destination to clique i are initially external to it. If in addition the clique i is a leaf, then,

broadcasting all items from super clique to the clique i requires $D_3 = 3(k-1)\frac{n}{p}|HUB_{max}|\log_2(k)$ slots while the

stations involved (*i.e.* stations in cliques located on the path to clique i) remain awake for $d_3 = 2(k-1)\frac{n}{p}|HUB_{max}|$

slots at most because this operation is performed for the k-1 cliques that are not super clique.

Lemma 2

Globally, the broadcast phase of items leaving the super clique to the clique i requires

 $D = 6(k-1)\frac{n}{p}|HUB_{\max}|\log_2(k) + 3(k-1)\log_2(k) \text{ slots for its execution. Each node involved in this phase}$

remains awake for not more than $d = 4(k-1)\frac{n}{p}|HUB_{max}| + 2(k-1)$ slots.

Proof: During the broadcast phase of items to the super clique, $D_1 = 3(k-1)\frac{n}{p}|HUB_{\max}|\log_2(k)$ slots are

needed and each station involved stays awake for $d_1 = 2(k-1)\frac{n}{p}|HUB_{max}|$ slots. The communication path to be followed by items of clique i, doing by the *superclusterhead* requires $D_2 = 3(k-1)\log_2(k)$ slots, while each station involved stays awake for $d_2 = 2(k-1)$ slots. The downhill phase of items to the clique i requires $D_3 = 3(k-1)\frac{n}{p}|HUB_{max}|\log_2(k)$ slots and each station involved stays awake for $d_3 = 2(k-1)\frac{n}{p}|HUB_{max}|$ times slots. By adding all the previous times required by different sub-phase of this phase, the result is obtained and then, we have $D = D_1 + D_2 + D_3$ and $d = d_1 + d_2 + d_3$ which are respectively the total number of slot re-

2.5. Fourth Stage: Local Broadcasts in Cliques

quire by this phase and the total time that stations involved stay awake.

At the end of the third stage, items in destination to clique i are already in clique i. Our purpose here is to show how different items will reach their final destination. For this, we use the routing protocol in a one-hop WSN presented by *Nakano et al.* [6]. In the worst case, the clique contains $|HUB_{max}|$ stations and this number will help us to determine the necessary number of slots for this stage. The maximum number of items to route in a network's

clique in this step is $\frac{n}{p} |HUB_{\max}|$. Nakano et al. therefore shown in [2] that the routing permutation problem in one clique can be solved efficiently in a network with $|HUB_{\max}|$ stations, contained $\frac{n}{p} |HUB_{\max}|$ items in $E = 2d \frac{n}{p} |HUB_{\max}| - 2(|HUB_{\max}| - 1))$ slots. Each station involved remains awake for at most $e = 4d \frac{n}{p}$, where $d = \left[\frac{\log_2(|HUB_{\max}|)}{\log_2(\frac{n}{p})} \right].$

The pseudo-code of our protocol is as follows (see Figure 5).

Theorem 1

Let *p* sensors in a multi-hop sensor network (n, p) with *n* items pretitled on it. Without clustering broadcast slots, the permutation routing problem can be solved in

$$G = 2\frac{n}{p} |HUB_{\max}| (3k \log_2(k) - 3\log_2(k) + d) + (k - 3)HUB_{\max} + 6(2k - 1)\log_2(k) + 3 \text{ time slot in the worst}$$

case with no station being awake for more than $g = 4 \frac{n}{p} ((k-1)|HUB_{max}|+d) + 11k - 6$ slots where

$$d = \left[\frac{\log_2\left(\left|HUB_{\max}\right|\right)}{\log_2\left(\frac{n}{p}\right)}\right]$$

1 E	Begin										
2	First stage: run the maximum clique formations protocol in [4] to obtain cliques;										
3	Second stage: run the hierarchical clustering algorithm in [10] with $k = \frac{G'}{2} $ on the										
	dominating set, G' derived from the first stage;										
4	Third stage: routing the external items for each clique;										
5	Begin										
6	General scheduling cliques;										
7	Identifying and scheduling cliques that have items in direction of the clique i;										
8	Begin										
9	Phase 1: sending the list of members of the clique i to superclusterhead;										
10	Phase 2: Broadcasting the members' list of clique i to the other clique;										
11	Phase 3: Identifying the items in destination to clique i among differents cliques;										
12	Phase 4: Scheduling the cliques to transfert the items to the clique i;										
13	End										
14	Sending external items identified in the tree to the clique i using the										
cyclic reception technique;											
15	Begin										
16	Phase 1: broadcast of items destined to clique i to the super clique;										
17	Phase 2: broadcast of items destined to clique i from the super clique;										
18	End										
19	End										
20	Fourth stage: local broadcast in cliques;										



Proof: By summing the results of both lemmas, the time required for internal routing in each clique and the time required for the general scheduling of the cliques, we obtain the result. Indeed, G = C + D + E + F and g = c + d + e + f.

Theorem 2

The maximum number of items in any given clique at a time is $2\frac{n}{p}|HUB_{\text{max}}|$.

Proof: Let clique j located on the path connecting clique i and the super clique. Let $|HUB_{max}|$, the number of stations in clique i. Two situations may occur when cliquei's items arrive in clique j.

If the number of stations of the clique j is less than $|HUB_{max}|$, let N be this number, then, the number of items in

the clique j will be
$$\frac{n}{p} |HUB_{\max}| + \frac{n}{p} \times N = \frac{n}{p} (|HUB_{\max}| + N) < 2\frac{n}{p} |HUB_{\max}|.$$

If the number of stations of the clique j is equal to $|HUB_{max}|$, then, the number of items in clique j will be

 $\frac{n}{p} \left| HUB_{\max} \right| + \frac{n}{p} \left| HUB_{\max} \right| = 2 \frac{n}{p} \left| HUB_{\max} \right|, \text{ which completes the proof.}$

Let $|HUB_{min}|$, be the number of stations in the clique with minimum size, we have the following result: Theorem 3

Assuming $|HUB_{max}| < 2|HUB_{min}|$, each station of the network has a memory capacity in $O\left(3\frac{n}{p}\right)$.

Proof: The proof of this theorem is made as follows:

- 1) Before the running of the protocol, it is assume that each station has in his internal memory a total of $\frac{n}{n}$.
- 2) During the protocol process, the worst case occur when on the path from the super clique to the clique of maximum number of stations $|HUB_{max}|$, items have to pass throw the clique of minimum number of stations

 $|HUB_{\min}|$. The total number of items destinate to clique $|HUB_{\max}|$ is $\frac{n}{p}|HUB_{\max}|$. Thus, each station of the

clique $|HUB_{\min}|$ must store $\frac{\frac{n}{p}|HUB_{\max}|}{|HUB_{\min}|}$ items. According to our hypothesis $|HUB_{\max}| < 2|HUB_{\min}|$, we ob-

tain $\frac{\frac{n}{p}|HUB_{\max}|}{|HUB_{\min}|} < \frac{2\frac{n}{p}|HUB_{\min}|}{|HUB_{\min}|} = 2\frac{n}{p}$. Hence, it is concluded that the memory allocated for routing items in

different stations is at most $2\frac{n}{p}$.

To obtain the maximal memory capacity of each station in the network, we just have to sum the size of the memory for its own items $\left(\frac{n}{p}\right)$ and the memory size for routing items $\left(2\frac{n}{p}\right)$. Therefore we obtain $3\frac{n}{p}$.

3. Simulation Results

In this section, we present the simulation results that we achieved. These simulations were performed in a desktop (Core i7.32 Go RAM, Ubuntu 14.04 LTS) with NS-2.35 and Java Environments. Our main problem has been to establish suitable experimental conditions. We assume that nodes are static and they have the same transmission radius. The experiments take place in a geographic square area of side L. The presented curves are the average of 100 experiments. We made the common assumption that two nodes are neighbors if and only if their Euclidian distance is less than 1 km. The nodes are in a square of 2 km side. In our implementation, the MAC layer is managed in such a way that a node can only receive one message at a time with the number of items sets to 1000.

3.1. Evolution of Sensor's Energy

The energetic model we use is similar to the one in [13],

$$E = ET + ER = N \times n \left(e_t + e_{amp} \times d^2 \right) + e_r \times n$$

where ET and ER are respectively the energy used for the transmissions and the receptions of items in the network. The energy dissipated by the transmitter, the amplifier and the receiver are respectively expressed by e_t, e_{amp} and e_r . Moreover, d is the Euclidian distance between nodes, N is the energy parameter mitigation $(2 \le N \le 4)$ and n represents the number of items. Thus, based on this model, we value ET to 5J and ER to 4J with initial energy of sensors to 1000J and then, we have the **Figure 6**. This figure is the one which represents communication between two nodes. The curve's energy dissipate while transmitting items to that of the items sender, and the curve of the energy waste while receiving items is that of the receiver node.

3.2. Average Number of Cliques

Figure 7 represents the evolution of the average number of cliques based on the number of nodes of the network. We randomly generate a graph with the labels on the nodes. Then, we partition the previous graph using the protocol of *Sun et al.* [4] and therefore we get the number of cliques.

3.3. Comparing Our Protocol to That of Lakhlef et al. [7] and That of Bomgni et al. [3]

In the **Table 1**, we did a comparative study of our protocol to those presented by *Bomgni et al.* [8] and by *Lakhlef et al.* [9]. From this table, it is clear that the awaking time taken by our protocol is less than that presented in [8] and [9]. For instance, for p = 1000 and n = 1000000, the sensors stay awake during 18203881 slots in our protocol, while in the protocols of *Bomgni et al.* and *Lakhlef et al.*, the sensors stay awake during 19006435 and 46454931 slots respectively. In addition to the awaking time which is better for our protocol, on can noted that





Table 1. Comparing	the time used	by our protoco	ol to that of Lakhlef et	al. and that of Bomgni et	al.
---------------------------	---------------	----------------	--------------------------	---------------------------	-----

	$P = 100, k = 38, HUB_{max} = 4$		$P = 500, k = 46, HUB_{\text{max}} = 20$			$P = 1000, k = 48, HUB_{max} = 25$			
n –	1000	10,000	100,000	5000	50,000	500,000	10,000	100,000	1,000,000
Bomgni et al.	44,873	249,160	2,292,028	178,077	1,473,690	14,429,812	227,239	1,934,439	19,006,435
Lakhlef et al.	44,109	436,968	4,365,561	230,226	2,297,692	22,972,355	465,063	4,645,960	46,454,931
Our Protocol	23,120	217,281	2,157,869	139,944	1,381,221	13,791,664	184,183	1,822,567	18,203,881



Figure 7. Average number of cliques.

our protocol is executed in consideration of the different memory sizes of sensors' network that is about $O\left(3\frac{n}{p}\right)$, in contrast to those presented in [8] and [9] where the memory size of the sensors are infinite.

4. Conclusion and Open Problems

We have proposed a hierarchical permutation routing protocol that reduces the workload of clusters heads. First, we divided the network in disjoint cliques using the protocol of *Sun* et *al.* [4]. Secondly, we transformed the previous disjoint cliques into a hierarchy of clusters by using the protocol of *Banerjee et al.* [10]. We finally broadcasted the items efficiently to their final destination. Contrary to the protocol presented in [8] and [9], where only CHs are responsible for ensuring the broadcast of external items of cliques, our protocol distributes this task to all the stations in the clique, and then reduces the workload of the CHs. Moreover, we showed that the number of items which are present in one clique at a time cannot exceed a certain limit, and the size of the memory of each station of the network is limited.

However, several open problems remain. In our future work, we plan to study fault tolerance, which guarantees that normal stations receive in a finite time, the items destined to them. We also, plan to secure this protocol to prevent malicious intrusions. It would also be interesting to see how to mitigate the constraint $|HUB_{max}| < 2|HUB_{min}|$.

References

- Lin, J. and Liao, M. (2010) A Clustering Patch Hierarchical Routing Protocol for Wireless Sensor Networks. *The 5th International Conference on Computer Science Education*, Hefei, 24-27 August 2010, 941-948. http://dx.doi.org/10.1109/iccse.2010.5593457
- [2] Al-Karaki, J.N. and Kamal, A.E. (2004) Routing Techniques in Wireless Sensor Networks. A Survey. *IEEE Commu*nications, 11, 6-28.
- [3] Bomgni, A.B. and Myoupo, J.F. (2010) An Energy-Efficient Clique-Based Geocast Algorithm for Dense Sensor Networks. Wireless Sensor Network, 2, 125-133.
- [4] Sun, K., Peng, P., Ning, P. and Wang, C. (2006) Secure Distributed Cluster Formation in Wireless Sensor Networks. 22nd Annual Computer Security Applications Conference, Las Vegas, 131-140. <u>http://dx.doi.org/10.1109/acsac.2006.46</u>
- [5] Pottie, G.J. and Kaiser, W.J. (2000) Wireless Integrated Networks Sensors. *Communications of the ACM*, 43, 51-58.
- [6] Nakano, K., Olariu, S. and Zomaya, A.Y. (2001) Energy-Efficient Permutation Routing in Radio Networks. *IEEE Transactions on Parallel and Distributed Systems*, 12, 544-557. <u>http://dx.doi.org/10.1109/71.932709</u>
- [7] Datta, A. and Zomaya, A.Y. (2004) New Energie-Efficient Permutation Routing Protocol for Single-Hop Radio Net-

works. *IEEE Transactions on Parallel and Distributed Systems*, **15**, 331-338. http://dx.doi.org/10.1109/TPDS.2004.1271182

- [8] Bomgni, A.B. and Myoupo, J.F. (2010) A Deterministic Protocol for Permutation Routing in Dense Multi-Hop Sensor Networks. Wireless Sensor Network, 2, 293-299. <u>http://dx.doi.org/10.4236/wsn.2010.24040</u>
- [9] Lakhlef, H., Bomgni, A.B. and Myoupo, J.F. (2011) An Efficient Permutation Routing Protocol in Multi-Hop Wireless Sensor Networks. *International Journal of Advancements in Computing Technology*, **3**, 125-133.
- [10] Banerjee, S. and Khuller, S. (2001) A Clustering Scheme for Hierarchical Control in Multi-Hop Wireless Networks. Proceedings of the 20th IEEE International Conference on Computer Communications, 3, 1028-1037. http://dx.doi.org/10.1109/infcom.2001.916296
- [11] Wadaa, A., Olariu, S., Wilson, L., Eltoweissy, M. and Jones, K. (2005) Training a Wireless Sensor Network. *Mobile Networks and Applications*, 10, 151-168. <u>http://dx.doi.org/10.1023/B:MONE.0000048552.15853.c2</u>
- [12] Heinzelman, W.R., Chandrakasan, A. and Balakrishnan, H. (2000) Energy-Efficient Communication Protocol for Wireless Microsensor Networks. *IEEE Aerospace Conference Proceedings*, 3005-3014.
- [13] Lindsey, S. and Raghavendra, C. (2002) PEGASIS: Power-Efficient Gathering in Sensor Information Systems. Proceedings of the 33th IEEE Hawii International Conference on Systems, 3, 1125-1130. http://dx.doi.org/10.1109/aero.2002.1035242
- [14] Cui, S. and Ferens, K. (2011) Energy Efficient Clustering Algorithms for Wireless Sensor Networks. The 2011 International Conference on Wireless Networks, Monte Carlo Resort, Las Vegas, 18-21 July 2011.
- [15] Raghunandan, G.H. and Lakshmi, B.N. (2011) A Comparative Analysis of Routing Techniques for Wireless Sensor Networks. *National Conference on Innovations in Emerging Technology*, February 2011, 17-22.
- [16] Basagni, S. (1999) Distributed Clustering for Multi Hop Wireless Network. *Proceedings of the IEEE International Symposium on Wireless Communications*, June 1999, 41-42.
- [17] McDonald, A.B. and Zanati, A. (1999) A Mobility-Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 17, 1466-1487. <u>http://dx.doi.org/10.1109/49.780353</u>
- [18] Amis, A., Prakash, R., Vuong, T. and Huynh, D. (1999) Max-Min D-Cluster Formation in Wireless Ad Hoc Networks. INFOCOM, 1, 32-41.
- [19] Baker, D., Ephremides, A. and Flynn, J. (1984) The Design and Simulation of a Mobile Radio Network with Distributed Control. *IEEE Journal on Selected Areas in Communications*, 2, 226-237. http://dx.doi.org/10.1109/JSAC.1984.1146043
- [20] Younis, O. and Fahmy, S. (2004) Distributed Clustering in Ad-Hoc Sensor Networks: A Hybrid, Energy-Efficient Approach. INFOCOM, 1. <u>http://dx.doi.org/10.1109/INFCOM.2004.1354534</u>