International Journal of Advanced Computer Science, Vol. 3, No. 5, Pp. 249-258, May, 2013.

# Heterogeneous Clustering for Secure, Energy-efficient and Fault tolerant Permutation Routing in Wireless Sensor Networks

Sébastien Faye, Jean Frédéric Myoupo, and Alain Bertrand Bomgni

**Manuscript**
***Received:*** *14,Dec., 2012*
***Revised:*** *25,Dec., 2012*
***Accepted:*** *9,Apr., 2012*
***Published:*** *15,Apr., 2012*

**Keywords**
***Wireless Sensor Networks, Clustering, Security, Energy Efficiency, Permutation Routing.***

***Abstract*— In wireless sensor networks (WSNs), security and economy of energy are two important and necessary aspects to consider. Particularly, security helps to ensure that such a network is not subject to attacks that involve reading, modification or destruction of information. This paper presents a protocol for permutation routing, which is secure, fault tolerant and energy-efficient. The proposed protocol is based on two main principles. First, the use of a heterogeneous hierarchical clustered structure to assign the most important roles to the sensors having the most energy, in order to ensure the protection and routing of data items. Second, the use of multiple processes based on this structure to ensure, regardless of network and sensors status, no data is lost and that a data item from a point A to a point B always arrives safety. This is the first protocol that provides such a QoS per permutation routing.**

## 1. Introduction

Wireless sensor networks (WSN) are from the family of mobile ad-hoc (MANET), but have additional features and constraints: typically, they consist of a wide range of sensors with limited energy capacity. Each sensor is powered by a non-rechargeable and non-replaceable battery ([3, 13]) and has a low capacity in terms of memory, calculation (CPU) and transmission range. Each sensor is able to harvest a set of data in a certain environment, and transmit it in multi-hop way to a base station (BS, also called sink node), which may act as the network instructor. The use of these networks is widespread in many applications. For example we can mention the monitoring of forests, critical infrastructures, or the detection of biochemical agents in military industries. Some examples of works can be found through [1], [3], [14], [33].

In such a network, the security is a crucial point that we need to study and put forward. In fact, WSNs have many constraints such as the communication medium, which is wireless: nowadays it is very easy to read, intercept and even modify the transmitted data and to compromise an entire network. Let us add to these inconveniences the sensors application context, which are usually deployed in hostile environments. In consequence, there is a need to secure the protocols, in order to guarantee authentication, exchanges confidentiality ([22, 32]), data integrity and network availability. In the literature, several security protocols have been proposed. We can mention TinySec ([19]) or μTESLA ([31]), which ensures the authentication of the packets sent from a BS to the whole of nodes (broadcast or multicast). In short, a good security system should be able to avoid external attacks (coming from an attacker from outside the network) as well as the internal attacks (from an internal attacker of the network, by compromising a node).

The technology related to sensor networks advancing day by day, it is common to see WSNs composed of several thousand units ([15], [37]). In large networks, sensors can be grouped into clusters based on their proximity to offer a better management and data transmissions in order to significantly increase the scalability, economy of energy, routing and consequently the lifetime of the network (eg. [11], [24], [25], [34]). To maintain consistency, a minimal hierarchy is created in each cluster, where the members agree on a chief: a cluster-head (CH for short), which is responsible for managing all members of its cluster and to carry out outwards operation.

### A. Statement of the permutation problem

Consider a wireless sensor network of $p$ stations with $n$ items pretitled on it (WSN(n, p) for short). Each item has a unique destination, which is one of the p stations. Each station has a local memory of size $O(n/p)$ in which $n/p$ items are stored. It is important to note that in general, some of the items stored in a station, say $i$, have not $i$ as final destination station. On the one hand, it can happen that none of these items belongs to it. On the other hand, the situation in which initially all items in $i$ belong to $i$ can also occur. The permutation routing problem is to route the items in such a way that for all $i$, $1 \leq i \leq p$, station $i$ contains all its own items at final.

*Sébastien Faye, Jean Frédéric Myoupo and Alain Bertrand Bomgni are with the Université de Picardie-Jules Verne, UFR Sciences, Labo. MIS, 33 rue Saint Leu, 80039 Amiens France. (faye@telecom-paristech.fr, jean-frederic.myoupo@u-picardie.fr✉, alain-bertrand.bomgni@u-picardie.fr).*

*B. Previous Work*

The number of studies specifically targeted to permutation routing in single hop wireless networks has grown significantly: it is shown in [27] that the permutation routing of n items pretitled on wireless sensor network of p stations and q channels with q < p, can be carried out efficiently if $q < \sqrt{\frac{n}{p}}$ . In [26], Myoupo solved the problem showing how the above restriction can be left. Datta in [7] derived a fault tolerant permutation routing protocol of n items pretitled on mobile Ad-hoc network of p stations and *q* channels (MANET(n, p, q) for short). He also assumed that $q < \sqrt{\frac{n}{p}}$ and in the presence of faulty stations some data items are lost. We came out with our work in [17] presenting a fault tolerant protocol, which avoids the loss of items. The first energy-efficient permutation routing appeared in [28]. A more efficient energy-efficient permutation routing protocol was presented in [8]. In [38] Walls et al. propose an optimal permutation routing on mesh networks. Another approach as an application of an initialization algorithm appeared in [17]. All these approaches assume that the WSN is a single hop network. Our former work in [18] presents a randomized algorithm for the same problem in multi-hop network with high probability. In our former work in [20] we show that the permutation routing problem can be solved in $3n + 6\log_2 k$ in the worst case, where *k* is the number of cliques after clustering.

*C. Basic Definitions*

**Definition 1**. A WSN is a set S of *n* radio transceivers or sensors, which can transmit and/or receive messages from each other. The time is assumed to be slotted and all sensors have a local clock that keeps synchronous time. In any time slot, a sensor can tune into one channel and/or broadcast on at most one channel. A broadcast operation involves a data packet whose length is such that the broadcast operation can be completed within one time slot. Also, all the communications are performed at time slots boundaries i.e. the duration operation is assumed to be sufficiently short. So, in the WSN with Collision Detection (CD for short), the status of an n-station WSN channel is:

- ***NULL*** : if no station broadcasts on the channel in the current slot.
- ***SINGLE*** **:** if exactly one station broadcasts on the channel.
- ***COLLISION*** : if two or more stations broadcast on the channel in the current time slot.

**Definition 2**. All communications are over wireless links. A wireless link can be established between a pair of nodes only if they were within wireless range of each other. Two sensors that have a wireless link will be said to be 1-wireless hop away from each other. They are also called neighbors. Moreover, each sensor belonging to a cluster is a resident of that cluster. Hence, this sensor may, in a given time unit, broadcasts a message to its neighbours.

**Definition 3.** Let us consider *p* stations *1, 2,...,* p which communicate in a multi-hop wireless sensor network WSN(n, p). We suppose that we have *n* items in the system. Then each station of a WSN (n, p) is assumed to have a local memory of size at least *O(n/p).*

**Definition 4.** We assume that the n items denoted $a_1$, $a_2$, ..., $a_n$ are pretitled on a WSN(n, p) such that for every *i*, $1 \le i \le p$, station *i* stores the *n/p* items. Each item has a unique destination station. It is important to note that hereafter a station knows the destination of items it holds. In fact the data item it holds is a couple (a(v), v). Where a(v) is the real data item belonging to sensor v. For every v, $1 \le v \le p$, let $h_v$ be the set of items whose destination is sensor v.

**Definition 5**. The permutation routing problem is to route the items in such a way that for all v, $1 \le v \le p$, sensor v contains all the items in $h_v$ . Consequently, each $h_v$ must contain exactly *n/p* items.

*D. Our Contribution*

We consider a WSN(n, p) with n items, p stations. We first propose to partition the network into single-hop clusters also named *cliques*. Secondly, we run a local permutation routing to broadcast items to their local destinations in each clique. Next, based on previous work ([9]), we partition the cluster heads of cliques with the hierarchical clustering technique. We show how the outgoing items can be routed to their destination cliques. We give an estimation of the upper bound of the number of broadcast rounds in the worst case.

The rest of this work is organized as follows. Some definitions and the environment considered in this work are presented in section 2. In section 3, we present some useful preliminaries. The permutation routing is described in section 4 followed by the simulation results in section 5. A conclusion ends the paper.

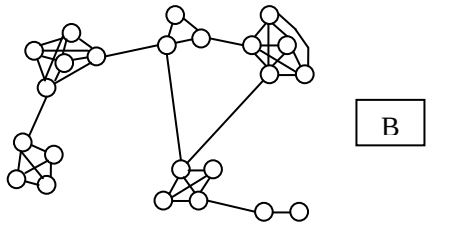
# 2. Preliminaries on Clustering

In order to better apprehend the next parts and in particular our formation protocol in 3.1, this section details two fundamental aspects which we will use: first Sun et al. cliques clustering [40], then the concept of virtual architecture suggested by Wadaa et al. [39].

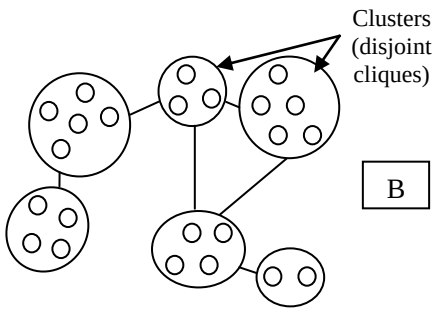
*A. A Secure Clustering Protocol*

As introduced earlier, nowadays it is common to find networks with many sensors, so there is a need to group those, using clusters. There are many works allowing their creation (example: [11], [24], [25], [34]) and are generally divided into two main families. On the one hand, leader-first protocols which first manage to elect a CH and to form clusters around (examples: LEACH [11], TEEN [24], APTEEN [25]). On the other hand cluster-first protocols, which first form the clusters, then elect a CH in each one. The retained solution is the latter and uses the protocol of Sun et al. ([34]). Our choice referred to this protocol because it is secure (i.e. able to avoid a majority of internal and external attacks). And also a protocol cluster-first is better in the context of what we want to make:

if changes were to happen on the network at different CHs, we would just make a re-election. There should be no need to rebuild everything unlike first-leader protocol. Lastly, a major advantage of this protocol is that it is based on disjoint cliques formed by the representative graph of the WSN. This ensures us that inside each cluster each member can reach another member in only one hop, which reduces considerably the cost of certain communications. Figure1. a. represents an example of network, while figure 1.b. represents the application of Sun et al. protocol.




(a)




(b)

Fig. 1. (a) Network example (b) Sun et al. formation.

### *B. Virtual Architecture*

Consider a randomly gathered network of sensors around BS, which has the ability to transmit information to some powers (to the most distant sensor) and one-way (at certain angles). The concept of virtual architecture which matters to us is the one developed by Wadaa *et al.* ([36]): the problem is that initially a node or set of nodes are not directly detectable by BS in space, no structure was clearly defined. The proposed solution therefore consists of a partition of the network into different zones (or areas) by BS. The latter has the possibility of disseminating information with more or less great range, this being used to create coronas; also, it has the possibility of disseminating information in certain directions, which is used by [36] in order to create various angular sectors. Zone (i, j) is the intersection of an angular sector j and a corona i. The sensors of the same area are therefore in the same geographical location and form a cluster. This is illustrated in Figure 2.

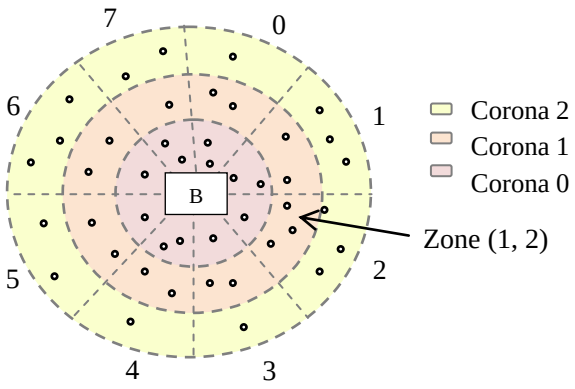



Fig. 2. Virtual architecture.

## 3. Heterogeneous Hierarchical Clustered WSN

Based on previous work ([9]), the aim of this section is to derive a heterogeneous hierarchical clustering by superposing the two clustering of subsection 2 in a cascading way.

### *A. Notations, Assumptions and Security Issues*

#### 1) *Notations*:

To clarify the continuation of our paper, we use the notations below. Some are not specified here but directly in the paper.

- [a-z]: Indicates a sensor.
- $CH^N$: Cluster-head of level N (introduced into 3. 1. 2).
- $ID_u$: single identifying of 4 bytes corresponding to the node U.
- $W^N\backslash\{w\}$ : the whole of the nodes present in the cluster of level N of the node w (without w).
- WSN*: the whole of the sensors of the network.
- a*: zero, one or more noted nodes a.
- a,b: a concatenate to b.
- D: a message to be transmitted.
- K{n/u}: a one way keys chain of size *n+1* generated by the node u.
- Ku, v: a secret key shared between a node u and a node v.
- Kbs,u: a secret key shared between the base station and a node u.
- MACK(M): an authentication message of 8 bytes generated over M by using the key K.
- H: a one way hash function (µTESLA).

#### *2) Model of Architecture:*

Our contribution is based on a layered clustering model. We yield a cascading protocol ([5]) over the clustered WSN (using [34] first) with the goal to get many layers (or levels) of clustering. At the initial level, the nodes are partitioned into clusters called “of level 1” and in each one of these clusters, a chief is being elected (a CH1). The clusters of level 1 are in their turn partitioned into clusters “of level 2”, and a chief would be elected for each cluster (a CH2 elected

among the CH1s) and so till level N where the cluster head CHN+1 of each cluster of level N elected among the CHNs. The election of CH can be conducted with energy factors: the CH having the highest layer is those with the most energy, as this formation is illustrated in Figure 3.

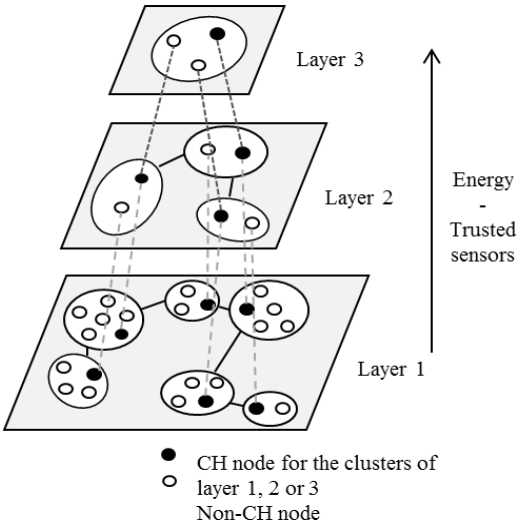



Fig. 3. An example of proposed model.

If the advantageous distribution of the network is tempting, the challenge here is a security question: it is necessary for us to make sure that a permutation routing is correctly carried out, without letting the network become an easy prey for the attackers and external threats.

*3) Assumptions:*

(i). WSN is static, i.e. composed of immobile nodes once deployed. Optionally, adding and deleting nodes (for fault tolerance) is allowed, but considered rare.

(ii). The base station (BS) is the only trusted entity in the network (it cannot be compromised), which has a strong energetic capacity than other nodes, and whose transmission range can cover the entire network. BS may also perform unidirectional transmissions (at certain angles).

(iii). We suppose that each node shares a secret key with BS, which is charged before deployment. Also, each node is loaded with the necessary cryptographic material key establishment with its neighbors (using mechanisms as proposed in [4, 5, 6], [35]).

(iv). Each sensor is capable of locating itself in space using GPS, triangulation or a system of positioning for ad hoc networks (eg: APS [29]).

(v) We also consider the following properties: each node knows its neighbors at 1-hop and has a unique identifier. A message sent by a normal node can be received correctly by all its neighbors (1-hop). All the messages exchanged between two nodes are authenticated, thanks to the key shared between these nodes. Each node can generate a public key based on a signature (realist assumption compared to the literature: [10], [23]). The messages (broadcast) are authenticated thanks to a combination of signatures and μTESLA protocol. The signature is used for the nonrepudiation of the data. The protocol μTESLA is used for an effective authentication of broadcast. The clocks of the nodes are synchronized, as μTESLA requires it. The keys distributed by the various nodes and the base station are authenticated (use of μTESLA or a certificate in order to ensure the authenticity of a received key). Finally, let us recall that a WSN is always connected.

### *B. Heterogeneous Clusters Formation Protocol*

We now present our cluster formation protocol, which is necessary to obtain the structure that we have just described. The latter is divided into four main phases. First the initialization which is orchestrated by BS in order to set up on the one hand the cryptographic material necessary to the basic security of the network and on the other hand the various identifiers of the nodes. Next we use an existing and reliable protocol in order to build our first cluster level. This minimal structure is necessary to install an additional mechanism of keys. We end up using a virtual architecture concept in order to form the next levels.

*1) Phase 1: Initialization:*

This phase occurs before the network deployment. The base station first generates a chain of keys K{n/bs} needed to perform broadcasts to all authenticated sensors - in order to create our formation, or possibly for other operations: alerts, etc. -. It then charge each sensor u with a single identifying IDu, with a secret key Kbs,u shared with itself in order to ensure future unicast communications (to guarantee confidentiality and authentication), and the first key K{0/bs} of its chain of key, in order to carry out broadcasts on the whole network (we use μTESLA use: to guarantee authentication). AT last, BS charges each node u with the cryptographic material key establishment with all its neighbors, for secure communications between pair of neighbor. Two neighboring nodes u and v has a shared key Ku,v.

*2) Phase 2: First Level Cluster Construction: Clustering in Cliques:*

As indicated in 2. A., here we choose a secure cluster-first protocol in order to build in all serenity our first level and to ensure us of his solidity. This second phase is thus initially the application the Sun et al. protocol ([34]). The latter uses the keys Ku,v set up between two nodes u and v, it also uses the authentication broadcast with μTESLA ([31]): each node u generates – using its cryptographic material – a chain of keys K{n/u} and distributes the first key of the chain K{0/u} to its neighbors.

Let us notice that, this protocol does not take into account the typical example of the multiple identities (sybil) or wormhole attacks. However, these attacks can be detected by using known techniques of the literature as the work of Y. Hu et al. for sybil attack ([30]) and the work of B. Parno *et al.* for wormhole attack ([12]).

At this stage we assume that:

- Each clique contains β sensor nodes and that there are at most δ (δ<β) faulty sensor nodes in a clique. Each sensor node has a local memory of size *2n/p*.
- Each node has many power levels with the crucial level, say $EC_l(i)$ to mean the crucial power level of sensor node i. $EC_l(i)$ is the necessary power for i to broadcast all the data items it detains in its clique. After broadcasting all its data items, it becomes faulty. In clear when the power level of node i reaches $EC_l(i)$, i broadcasts an alert signal to has a help for saving its data. Then it broadcasts all its data items in its clique. The broadcasted items are saved successively beginning by the node of the lowest identity in the clique to the one of the higher identity. It is important to note that a sensor j that has saved the data items of a faulty node j', will play the role of j' in the future.
- Two nodes of the same clique cannot broadcast theirs alerts at the same time.

Once clusters created, the nodes inside each one agree on a chief and proceed to an election: we obtain a $CH^1$ (cluster-head of level 1) in each cluster of level 1(Note the $CH^1$ of clique i). We assume that the elected nodes are those with the higher power level, in order to protect the cluster management and future builds. Finally, each elected CH sends a message to BS containing the list of members of its cluster. The BS being informed by the network members, he is able to launch the following phase when it receives all CH acknowledgments.

*3) Phase 3 (Recursive): Higher Levels Construction:*

The preceding phase enabled us to obtain a really healthy base for each cluster of level 1. Now, as described into 2. 2. 2., we rely on a virtual architecture mechanism similar to [36] to partition our level 1 clusters at higher levels. It's BS - only trusted entity in the network - which is in charge of this operation.

Initially, BS knows the network, and determines - according to the number of nodes and the will of partitioning fixed by the administrator - a range coefficient Cp (between 0.1 and 1 - 1 representative 100% of the distance separating BS of the most distant sensor in the network - this parameter can be given by successive BS broadcasts at the time of initialization) and an angular coefficient Ca (between 1° and 360°). Everything then depends on the system administrator wishes: to make a lot of levels, we use a low Ca and Cp, to make a minimum we use a larger Ca and Cp. Other calculations related to virtual architecture are not detailed in this paper because are already fully the subjects of a study in [36]. BS thus is able to cut the network by making broadcasts to different ranges and angles, according to Ca and Cp, as it is suggested in figure 4.

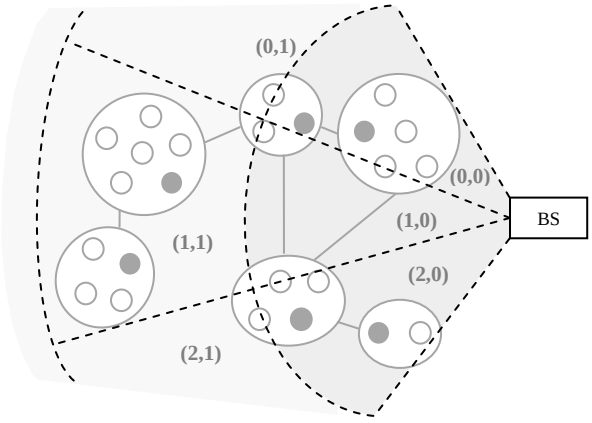



Fig. 4. BS network cutting with Cp = 0.5 and Ca = 40°. Here, we take the second level formation case. Broadcast step.

Each zone defined by BS is indicated by a couple of integer: (angular number, corona number). The cluster generation process then is as follows for each level N (N > 1):

*Step 1*. BS performs a broadcast using the key K{n/bs} in order to communicate in an authenticated way the couple of integer to all nodes. BS thus successively broadcasts a message of the type (angle, corona) according to Ca and Cp and thus to various angles and coronas (eg on figure 4).
For each zone (angle, corona) defined by BS according to Ca and Cp:

$$BS \rightarrow WSN^* : D, MAC_{K\{j/bs\}}(D), K\{j/bs\}$$

With K{j/bs} the current key of the key chain K{n/bs}, D the integer couple to be sent and corresponding to a certain zone according to BS.

*Step 2*. Each node u then receives the message. It first authenticates the - revealed - key K{j/bs} by using the previously stored key K{j-1/bs}: for that it uses the H irreversible hash function it holds, and checks the correspondence K{j-1/bs} = H(K{j/bs}). Once this first stage done, the node checks the authentication provided by MAC attached to the message and is able to upgrade its last known key. This is a simple application of the protocol μTESLA. Finally, a node is informed of the parameter (angle, corona) which is affected for him. Each node w ϵ $CH^1$ then communicates to all members of its level 1 cluster the parameter that it holds, by using the key chain K{j/w} of K{n/w} for authentication (broadcast with μTESLA). The goal is to set in agreement all the members of each cluster of level 1 (in case some members have a different setting).

$$w \rightarrow W^1 \backslash \{w\} : D, MAC_{K\{j/w\}}(D), K\{j/w\}$$

*Step 3*. They then read the parameter and upgrade their local value if there is a difference with the value broadcasted by BS, and return an acknowledgment containing this end value to their $CH^1$, which is authenticated with Ku,bs.

*Step 4*. Each $CH^1$ thus receives a group of answers that it cannot read or modify. Once all the responses received, it sends them all to BS, with the signature Kch,bs, by including IDu of members which have not responded.

*Step 5.* BS receives a message, authenticates it, and then checks one by one all the clusters nodes acknowledgments. If an inconsistency is noticed, or if it does not receive a message from a CH, it takes measures: re-election, banishment of nodes, or else.

*Step 6.* At this time, clusters of level N are implicitly formed: a cluster of level N is a set of clusters of level N-1, whose members have the same parameters (angle, corona) and are directly linked.

*Step 7.* The continuation of the algorithm then consists of electing a $CH^{N+1}$ among all the accessible $CH^N$ between them without changing the parameter (angle, corona), like illustrates it figure 4. Therefore a $CH^{N+1}$ is also $CH^N$, $CH^{N-1}$, $CH^{N-2}$,… $CH^1$. The cluster is considered to be formed. The cluster concept is a little bit abstract here, because a node belonging to a cluster of level N (N > 1) does not directly know all the other members, it has only knowledge of its $CH^N$ and members of its level 1 cluster. Here, we do not detail the election procedure itself, which depends on the parameters desired by the user (according to the supplied energy, of the identifier, etc.).

*Step 8.* Once the election made, each lately elected CH informs BS which starts again this entire phase for a higher level, by multiplying Cp and Ca by a factor i to fix. Cluster formation is complete when BS sees that there is nothing left but one CH for the level N.

Cluster formation is completed and is not to be re-run. However it is possible that adjustment operations are performed internally, such as update operations: fault tolerance, adding nodes, or yet banishment mechanisms or re-election if a malicious node is detected.

# 4. Permutation Routing Protocol

Once we establish our heterogeneous multi-level clustered structure, permutation routing can be performed at various times during the life of the network. Note that a $CH^1$ is connected to the BS via $CH^2$, $CH^3$,…, $CH^N$. The BS, being the only trusted entity in the network, is used solely for the purpose of securing information that would be passed without having to directly manage. Clearly, the BS is responsible for disseminating the different keys that enable us to provide security information to be exchanged between all nodes.

We recall that we have *p* sensors and *n* data items pretitled in these *p* sensors. Hence each sensor has a locale memory of size $O(n/p)$. The time is slotted. Our Approach to provide permutation routing in multi-hop sensor network consists of the following five phases:

### A. Phase 1: Key Distribution

The BS is a trusted entity. The BS generates two private keys, K1 and K2, which it keeps and refers to all nodes in the network; it releases K1 to all nodes that are non-CH nodes, and it releases K2 to all CH nodes. This exchange must be done in a confidential and authenticated way, for example, by using the key $K_{bs,u}$ to provide a key from the BS to a node u. Another solution would be to use a delivery mechanism, such as μTESLA, allied to an aspect of ensuring confidentiality [31]. We assume here that once the keys are held by different nodes, frauds are detectable: if a malicious CH were to transmit its key K2 to a compromised member of its cluster, the fraudulent transaction would be detected by a third member of the clique (see security analysis in the next section). Finally, we raise the specific case of a cluster at level 1 consisting of a single member. In this case, the CH is informed of the keys K1 and K2 to manage transactions that are usually carried out only by two types of nodes in a cluster.

### B. Phase 2: Clustering in cliques

Run the *secure Clustering in cliques* Procedure of the subsection 3.2.2. Assume that this procedure yields k cliques, thus k cluster-heads (i.e., $CH_{clique\text{-}I}$, $1 \le i \le k$). Next we will only focus on these cluster-heads and consider a network, say G', whose sensors reduce to these k cluster-heads.

### C. Phase 3: Local Broadcasts in cliques

The idea of this phase is similar to the single-channel-routing protocol in [28]. The broadcast item here is a couple (a(v), v) where a(v) is the data item belonging to sensor v. It can be summarized as follows. $CH^i$ invites each node of its clique to broadcast one by one the data items it holds. In each slot, the sensor whose identity matches the destination of the item being broadcast copies the item in its local memory. If no sensor of the clique is the destination of the broadcast data item then the *sensor broadcasting keeps this data item and counts the number of these its outgoing data items*. Note that the cluster head has the IDs of all the residents of its cluster. The broadcasts are carried out on cliques. So the clique with the great number of sensors (say $Clique_{Max}$) should help to estimate the total broadcast rounds of this phase. In terms of security, the keys are now known from the various sensors inside clusters of level 1. For each cluster of level 1: each member sends to other members of its cluster, for example, $*W^1$, the data that it holds. This can be expressed as:

$$u \rightarrow W^1 : D, MAC_{Ku,j}(Data), K_{u,j}$$

$W^1$ are all sensors of level 1. $K_{u,j}$ is the key shared by u and each sensor j belonging to $W^1$.

At the end of this phase all data items that do not belong to the sensors of a clique are saved in the clique. The goal now is to route them to their final destinations.

***Lemma 1****: Since $Clique_{Max}$ detains the maximum number of sensors, it should need the maximum number of broadcast rounds in this phase. Therefore this phase needs $(n/p)/Clique_{Max}/+p^2-p$ time slots with no station being awake for more than $(2n/p^2)/Clique_{Max}/+2p$ time slots according to [28].*

### D. Phase 4: Higher levels secure Clustering

As in in subsection 3.2.3 we derive recursively the higher levels clusters over G' (the network which nodes are the cluster-heads of the cliques) i.e., over $CH^i$, $1 \le i \le k$. The

resulting higher level cluster-head, say $CH^N$ knows all the residents of its cluster, thus has their IDs named $CH^i$ in phase 1.

*E. Phase 5: Broadcasting outgoing data items*

Here we use of breath-first tree (BFS tree) created in the above sub-section on the hierarchical clustering. In this tree a node broadcasts simultaneously to its parents and children, and receives only from its parents or its children.

In this step the sensors whose cluster-head is $CH^i$ (the root of the tree) are the first to broadcast their out-going data items as follows:

(i). *Data encryption*: each node sends this encrypted result to its $CH^N$ by authenticating the message using the key $Ku,CH^i$: for each sensor u belonging to $W^i\{CH^i\}$ we have :

$u \rightarrow CH^i : K1(Data)||MAC_{Ku,ch\{1\}}(K1(Data))||K_{u,CH}{}^i)$

(ii). $CH^N$ invites its residents one by one to broadcast its outgoing-data items once every two slots. It collects this outgoing data and broadcasts it (in the next slot) in its turn to its sons on the BFS tree. The sons in their turn broadcast it till the leaves are reached.

(iii). Next the left most son of $CH^N$ on the BFS tree is invited by its father to proceed as in (i) above.

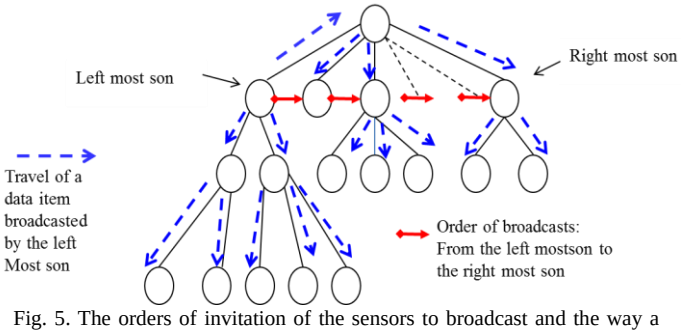



Fig. 5. The orders of invitation of the sensors to broadcast and the way a broadcasted data travels over the network.

Its sons and $CH^N$ collect the broadcasted data item. $CH^N$ broadcasts it to the rest of its sons and its broadcast spreads till the leaves. The same procedure is carried out by the sons of the left most son of $CH^N$. And so on till all the left leaves are reached. This procedure is depicted in Figure 5.

On receiving a data item, a $CH^i$ broadcasts it in its clique in the next slot. It is important to note that when a sensor of Clique i receives a data item it checks if it is its own. If it is the case it keeps it.

(iv). After the last data item coming from the leaves of the left tree has been broadcasted to the right tree by $CH^N$, this later informs its right son to initiate the same procedures as in (ii) above

***Lemma 2***. *$2n+6log_2k$ time slots are necessary in the worst case for the accomplishment of this phase.*

***Proof.*** In the worst case all data items detained by each sensor are outgoing items. Therefore all sensors have to broadcast these outgoing items one every 2 slots. Thus 2n time slots are necessary. The last item coming from a leaf of the left tree has to travel to the root, i.e., $2log_2k$ time slots at most. The last item coming from a leaf of the right tree has to travel to the root, and from the root to the leaves of the left tree i.e., $4\log_2 k$ time slots.

***Protocol_Secure_Energy-Efiicient_Fault-Tolerant Permut-Routing- muti-hop-Wsn***

**INPUT**: *wsn of p sensors in which a sensor may not have its own data items*
**OUPUT**: *wsn in which each sensor has its own data items*
**Begin**

1. Run the secure clustering clique protocol in [23].
2. In parallel on cliques, run the Energy Consumption Analysis protocol single-channel-routing of [27] for single hop wsn.
3. Run the cascading heterogeneous clustering algorithm on G' derived from 1 and derive $CH^2$, $CH^3$, …, $CH^n$ for each clique.
4. Broadcasts of outgoing data Items.

## 5. Security Analysis

We here study more in details the secure aspect of our protocol. We do not reconsider the heterogeneous cluster formation protocol in which both the unfolding and the security seem clear to us, but directly on the permutation routing protocol.

Concerning the local broadcasts in clique (phase 2), ensor nodes of a clique share a key, which ensures the authenticity of data received. It is detailed in phase 2. Our protocol guarantees that if there is an attempt to modify, delete, this is detectable by BS, the only trusted entity in the network. A majority of the external attacks (in a clique) are thus avoided, and the compromising

Broadcasting outgoing data items (Phase 4) is to route data items by simply going over the hierarchy of our structure, namely from a $CH^1$ to a $CH^N$. The authenticity of the package being guaranteed, it is impossible for a node j located between u and BS to modify packets contents. In the same way if such a node does not wish to transmit information that is forwarded to it, the node u is able to detect the error after a certain time (in slots). Our protocol guarantees that if there is an attempt to modify, delete, or change the path of a data, this is detectable by BS, the only trusted entity in the network.

A majority of the external attacks are thus avoided. However certain attacks are not managed by our protocol, it is the case, for example, of jamming attacks attempted on the whole network. For security and attacks on WSN the readers can find more details in [30, 31].

## 6. Fault Tolerance Analysis

We recall the Assumption of phase 1:

- Each clique contains β sensor nodes and that there are at most δ (δ<<β) faulty sensor nodes in a clique. Each sensor node has a local memory of size *2n/p.*
- Each node has many power levels with the crucial level, say $EC_l(i)$ to mean the crucial power level of

sensor node i. $EC_l(i)$ is the necessary power for i to broadcast all the data items it detains in its clique. After broadcasting all its data items, it becomes faulty. In clear When the power level of node i reaches $EC_l(i)$, i broadcasts an alert signal to has a help for saving its data. Then it broadcasts all its data items in its clique. The broadcasted items are saved successively beginning by the node of the lowest identity in the clique to the one of the higher identity. It is important to note that a sensor j that has saved the data items of a faulty node j', will play the role of j' in the future. Clearly j who inherited data items of j 'will play the role of j'. In this case there is no data loss as in [7].

# 7. Energy Consumption Analysis

Here, we use a model adopted by many efficient contributions ([11, 39] for example).

$$E = ET + ER = N \times (e_t + e_{amp} \times d^n) + N \times e_r \quad \text{(Equ. 1)}$$

Where ET and ER are the total energy used respectively in transmission on the network, and reception. In detail, N represents the number of nodes of the network, D the distance between the nodes, and N a parameter of energy attenuation ($2 \leq N \leq 4$). The energy used for the transmission is divided into energy for the radio transmission $e_t$ and the amplifier $e_{amp}$. The energy used for the reception is represented by unit (for each node) by $e_r$.

## A. Phase 1: Strategy of locating the CH in the Central Area of the Cluster

Power consumption can thus be studied under various levels, starting with our hierarchical formation. On the one hand, power consumption is less inside each cluster of level 1 (cliques), where each member has the possibility of communicating with another member in only one hop. Also, a CH1 can directly broadcast to every member of its clique in one hop. Consequently, the energy required for the formation of clusters of level 1 is less because based on cliques. This has a direct impact on the equation (1). On the other hand, during the formation of higher levels clusters, it is the BS that supports most of the actions needed to this formation, which ensures lower energy consumption over all network, while providing some security. Logically, BS is an entity with more energy than other sensors. This poses no problem. As before, this has an impact on the equation (1).

## B. Reducing power consumption during phase 3 (Higher levels secure Clustering)

Only clique cluster-heads and gateway nodes are involved in this scenario. They are the only sensors that are awake during the hierarchical clustering. The other sensors are asleep and will be waked up using the "Magic packet Technology" [2]. It is also known as the " Wake On Lan" (WOL for short). It consists in the ability to switch on remote computer through special network packets. Wol is based on the following principle: when a PC shuts down, the network card still gets power and keeps listening to the network for a magic packet to arrive. This technology was first designed for static wired networks, later a wireless version has been derived [21].

## C. Strategy of periodic hibernation

The study of energy consumption corresponding to our hierarchical formation being made, it remains for us to study phase 4 (broadcasting outgoing data items). For each data item it consists of two routings: an upward one from one or more $CH^1$ to $CH^N$, and a descent one from $CH^N$ to one or more $CH^1$. This routing is optimal in the sense that the path in which data items travel is simple, and does not consist in a flooding - even partial – of the network. Energy used is thus really minimized. Sensors awake only during intra cluster broadcasts, i.e. in Broadcast phase. In this phase the terms ET and ER of the formula (1) are minimized since all intra cluster broadcasts have sensors at one hop as destinations.

# 8. Simulations

In order to measure the effectiveness and to prove the flexibility of our formation protocol, we carried out some simulations, first described through figure 6. We successively took a population of 50, 150 and 300 clusters of level 1, as well as a whole of values for the coefficients Ca and Cp. For each possible case, we made 10 different simulations in order to evaluate an average location of the base station in the network, the distribution of different sensors, and the average number of clusters of level 2 that are possible to construct for such features. The results are visible on the graph in Figure 6 and allow us to have a view of the flexibility provided by our formation: everything is really a function of Ca and Cp values, which are chosen depending on the application to achieve and on the density of information to be aggregated. As we can see, the more we take a small value of Ca and Cp, the more the number of clusters of level 2 rises, which enables us to have a certain control of the number of our clusters of level 2. Concerning level 1, very suitable simulations were carried out in Sun et al. paper ([39]). Concerning the higher levels, the number of clusters depends on the multiplying coefficient on Ca and Cp, which we noted i. Let us take the case where i is equal to 2, and where Ca = 180 and Cp = 0.5 for level 2. Then for level 3 we have Ca = 360 and CP = 1. Thus level 3 has only one cluster.

Figure 7 shows the evolution and the comparison curves of the average number of broadcast rounds with respect to the number of sensors. It shows that there is a significant gap between the protocol in [3] and our protocol.

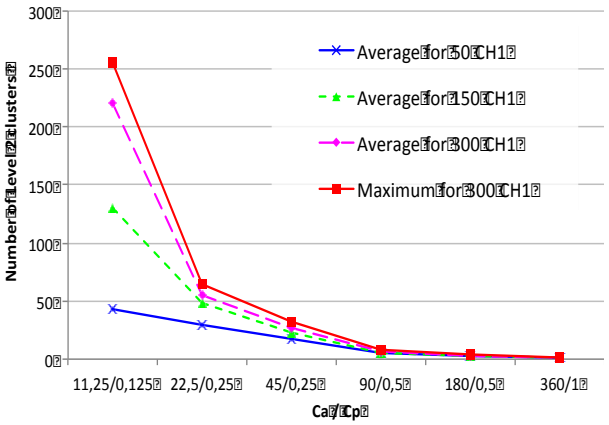



Fig. 6. Number of Level 2 cluster depending on Ca / Cp.

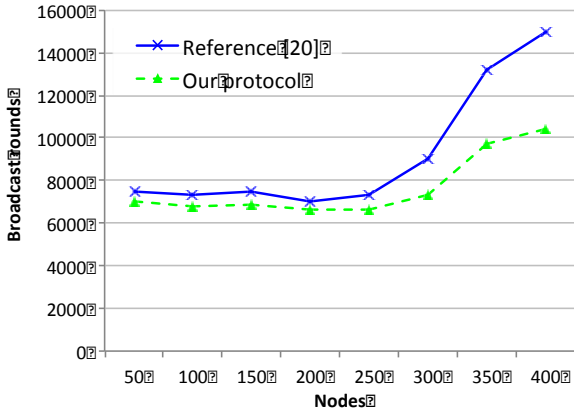



Fig. 7. Comparative curves of the average number of broadcast rounds.

## 9. Conclusion

The solution suggested through this document is a secure approach making it possible to carry out in a simple and fast manner the permutation routing in WSN. The hierarchical structure on which our protocol is based allows a distributed use of the network, and especially efficient use, for a control always ensured by BS. It avoids a majority of attacks [34]. Indeed, in addition to combining the essential aspect of security, our protocol is energy-efficient and uses a global structure with the network to reduce overhead, instead of local structures with certain regions (more constraining) that increase significantly the broadcast rounds overhead as it is the case for [20].

In future work, it would be interesting to study the problem by including certain nodes mobility in the network. Although fault tolerance and the addition of nodes are discussed here, the dynamics of the network are still very limited.

## Reference

[1] J. Agre and L. Clare, An integrated architecture for cooperative sensing networks, *IEEE Computer* 33(5) (2000) 106-108.

[2] AMD. White Paper; Magic Packet Technology, Nov 1995. http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/20213.pdf

[3] I. F. Akyildiz, W. Su and Y. Sankarasubramaniam. Wireless sensor networks: a survey, *Computer Networks* (38), pp. 393-422, 2002.

[4] R. Blom. An Optimal Class of Symmetric Key Generation. *Advances in Cryptography: EUROCRYPT 84, Lecture Notes in Computer Science*, 209, Springer-Verlag, Berlin, pp. 335-338, 1984.

[5] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung. Perfectly secure key distribution for dynamic conferences, *the 12th Annual International Cryptology Conference on Advances in Cryptology, Lecture Notes in Computer Science*, vol. 17, pp. 471-486, 1992.

[6] H. Chan, A. Perrig and D. Song. Random Key Predistibution Schemes for Sensor Networks. *IEEE Symposium on Security and Privacy*. Okland California USA, pp. 197-213, 2003.

[7] A. Datta. "Fault-tolerant and Energy-efficient Permuutation Routing Protocol for Wireless Networks*". 17th IEEE Intern. Parallel and Distributed Processing Symposium (IPDPS'03)*, Nice, France, 22-26, 2003.

[8] A. Datta, Albert Y. Zomaya: "An Energy-Efficient Permutation Routing Protocol for Single Hop Radio Networks". *IEEE Trans. Parallel Distrib. Syst.* 15(4): 331-338 (2004)

[9] S. Faye, J. F. Myoupo, "An Ultra Hierarchical Clustering-Based Secure Aggregation Protocol for Wireless Sensor Networks", *Advances in Information Sciences and Service Sciences(AISS)* Vol. 3, No. 9, pp. 309 ~ 319, 2011.

[10] N. Gura, A. Patel, and A. Wander. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. *The 2004 Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2004.

[11] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-Efficient Communication Protocol for Wireless Microsensor Networks, *the 33th IEEE Hawii International Conference on Systems*, pp. 3005-3014, 2000

[12] Y. Hu, A. Perrig, and D. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. *In INFOCOM*, April 2003.

[13] C. Intanagonwiwat, I. F. Akyildiz, W. Su and Y. Sankarasubramaniam. Wireless sensor networks: a survey, *Computer Networks* (38), pp. 393-422, 2002.

[14] C. Intanagonwiwat, R. Govindan and D. Estrin, Directed diffusion: A scalable and robust communication paradigm for sensor networks, in: *Proc. MOBICOM'00*, pp. 56-67, 2000.

[15] J.M Kahn, R.H Katz and K.S.J. Pister, Mobile networking for Smart Dust, in: Proc. *MOBICOM'99,* pp. 17-19 (1999).

[16] D. Karimou and J.F. Myoupo. "A Fault Tolerant Permutation Routing Algorithm in Mobile Ad Hoc Networks". *International Conference on Networks (ICN'05),* ), Part II, *LNCS 3421*, pp.107-115, 2005

[17] D. Karimou, J. F. Myoupo. "An Application of an Initialization Protocol to Permutation Routing in a Single-hop Mobile Ad-Hoc Networks". *Journal of Supercomputing* vol. 31, no3, pp. 215-226, 2005

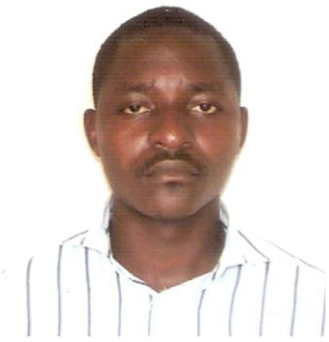
[18] D. Karimou et J. F. Myoupo: "Randomized Permutation Routing in Multi-hop Ad Hoc Networks with Unknown destinations". *IFIP International Federation of Information Processing*, vol. 212, p. 47-59 2006

[19] C. Karlof, N. Sastry and D. Wagner. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. *The 2nd international conference on Embedded networked sensor systems*, Vol. V, pp.. 162-175, 2004

[20] A. Bomgni et J. F. Myoupo, An Efficient Permutation Routing Protocol in Multi-Hop Wireless Sensor Networks. *International Journal of Advancements in Computing Technology (IJACT)* Vol. 3, Number 6, pp. 207-214, 2011
[21] J. Lewis, Wake On LAN over wireless. 2008. http://www.johnlewis.ie/2008/07/10/wake-on-lan-over-wireless/
[22] C. Li, Z. Wang, and C. Yang, Secure Routing for Wireless Mesh Networks, *International Journal of Network Security* Vol. 13, No. 2, pp. 109-120, 2011
[23] D. J. Malan, M.Welsh, and M. D. Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In *SECON*, pp. 71-80, October 2004.
[24] A. Manjeshwar and D. Agrawal. TEEN: A protocol for enhanced efficiency in WSN. *Proceedings of the 15th International Parallel & Distributed Processing Symposium*, pp. 2009-2015, April 23-27, 2001.
[25] A. Manjeshwar and D. Agrawal. APTEEN: A hybrid protocol for efficient routing and a comprehensive information retrieval in WSN. *Proceedings of the International Parallel and Distributed Processing Symposium*, pp. 195-202, April 15-19, 2002.
[26] J. F. Myoupo. "Concurrent Broadcasts-Based Permutation Routing algorithms in Radio Networks*". IEEE Symposium on Computers and Communications, (ISCC'03),* p.1272-1278, 2003.
[27] K. Nakano, S. Olariu and J.L. Schwing. "Broadcast-Efficient protocols for Mobile Radio Networks". *IEEE Trans. Parallel Distrib. Syst.*, vol.10, pp.12, 1276-1289, 1999.
[28] K. Nakano, S. Olariu, Albert Y. Zomaya: "Energy-Efficient Permutation Routing in Radio Networks". *IEEE Trans. Parallel Distrib. Syst.* 12(6): 544-557 (2001)
[29] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)," *Proceedings of IEEE Global Telecommunications Conference*, San Antonio, pp. 2926-2931, 2001
[30] V. Palanisamy, P. Annadurai, "Secure Geocast in Ad Hoc Network Using Multicasting Key Distribution Scheme (SGAMKDS)," *International Association of Computer Science and Information Technology - Spring Conference*, pp. 190-194, 2009
[31] B. Parno, A. Perrig, and V. Gligor. Distributed detection of node replication attacks in sensor networks. *In IEEE Symposium on Security and Privacy*, pp. 49-63, 2005.
[32] A. Perrig, R. Szewczyk, V. Wen, D. Cullar and J. D. Tygar. Spins: Security protocols for sensor networks, *In Proc. of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 189-199, 2001.
[33] I. A. Saroit, S. F. El-Zoghdy, and M. Matar, A Scalable and Distributed Security Protocol for Multicast Communications, *International Journal of Network Security*, Vol. 12, No. 2, pp. 61-74, 2011
[34] K. Sun, P. Peng, P. Ning and C. Wang. Secure distributed cluster formation in wireless sensor networks. *22nd Annual Computer Security Applications Conference*, pp. 131-140. 2006
[35] S-F. Tzeng, C-C. Lee, and T-C. Lin, A Novel Key Management Scheme for Dynamic Access Control in a Hierarchy, *International Journal of Network Security*, Vol. 12, No. 3, pp. 178-180, 2011
[36] A. Wadaa , S. Olariu , L. Wilson , M. Eltoweissy , K. Jones, Training a wireless sensor network, *Mobile Networks and Applications*, vol.10, pp.151-168, 2005.
[37] B. Warneke, M. Last, B. Leibowitz and K. Pister, SmartDust: communicating with a cubic-millimeter computer, *IEEE Computers* 34, pp. 44-51, 2001
[38] I. S. Walls and J. Žerovnik: Optimal permutation routing on mesh networks. *International Network Optimization Conference*, April 22-25, Spa, Belgium, 2008.
[39] D. Wei, S. Kaplan and H. A. Chan, Energy Efficient Clustering Algorithms for Wireless, Sensor Networks, *Proceedings of IEEE Conference on Communications*, Beijing, pp. 236-240, 2008.

**Sébastien Faye** was born in 1988 in France. He is currently a PhD student at the Computer Science and Networking Department (INFRES) of Telecom ParisTech, Paris, France. He obtained his Master degree in Computer Science from the university of Picardie Jules Verne in 2011. His current research interests include Intelligent Transportations and sensor networks.

**Jean Frédéric Myoupo** is Professor of computer science in the University of Picardie-Jules Verne, Amiens, France since 1994, where he heads the *parallel and Mobile computing.* He obtained the Ph. D degree in applied mathematics from the university of Toulouse 3, France in 1983, and the Habilitation in Computer in 1994 from the university of Paris 11, Orsay, France. Dr. Myoupo has served as member of program committee of international conferences as PDPTA, CIC, OPODIS, IPDPS workshops, HPCS, AP2PS, WINSYS, ICNS, ICWN, IEEE-RIVF, IEEE ISSPIT, ISCA-PDCS. Dr Myoupo is an Associate Editor of "*ISCA International Journal on Computers and their Applications"* and a Member of the editorial board of *"Studia Informatica Universalis '*.



**Alain Bertrand Bomgni** was born in Cameroon in 1981. He is currently a Ph. D student in computer Science at the university of Picardie Jules, Amiens, France. He obtained his Mater degree in computer science in 2006 from the university of Yaounde 1, Cameroon. His current research interests include parallel algorithms and architectures, Interconnection Networks.