Peer-to-Peer Netw. Appl. (2016) 9:640–655
DOI 10.1007/s12083-015-0360-0

CLEVER: Cluster-based Energy-aware Virtual Ring Routing in randomly deployed wireless sensor networks

Ghofrane Fersi[1] · Wassef Louati[1] · Maher Ben Jemaa[1]

Received: 30 September 2014 / Accepted: 20 April 2015 / Published online: 15 May 2015

Abstract Energy-aware routing is an important remedy to face the quick failure of energy-constrained nodes in Wireless Sensor Networks. Network clustering with electing energy-powerful nodes as cluster heads is a perfect solution. However, such clustering requires ideal nodes placement to afford best performances. Manual nodes placement is not always possible, the sensors can be randomly deployed. In such networks, the cluster heads cannot always communicate directly. In this paper, we present a novel clustering strategy for randomly deployed heterogeneous sensors, in which a cluster is defined as a set of energy-powerful nodes placed at the range of each other. The proposed protocol, called CLEVER (Cluster-based Energy-aware Virtual Ring Routing), uses virtual identity-based routing for intra and inter-cluster communications. The experimental results show that CLEVER increases drastically the network lifetime and optimizes efficiently sensors energy.

Keywords Wireless sensor networks · Distributed Hash Tables · Energy-aware routing · Clusters

✉ Ghofrane Fersi
ghofrane.fersi@redcad.org

Wassef Louati
wassef.louati@redcad.org

Maher Ben Jemaa
maher.benjemaa@enis.rnu.tn

[1] Research Laboratory of Development and Control
of Distributed Applications (ReDCAD), Department
of Computer Science and Applied Mathematics,
National School of Engineers of Sfax, University of Sfax,
BP 1173-3038 Sfax, Tunisia

1 Introduction

Wireless Sensor Networks (WSNs) are made up of a large number of sensors that have sensing, processing and communication functionalities [1, 2]. The main role of these sensors is to collaborate with each other in order to supervise given phenomena and to send results to base stations. WSNs are facing a lot of challenges such as energy optimization, fault tolerance, network coverage, security.

Energy optimization remains one of the most important challenges since it is directly related to the network lifetime. A lot of researches [3] have been proposed in order to face this challenge. Most of these researches, assume that the network is homogeneous, that is all sensor nodes have the same sensing and processing capabilities and the same initial amount of energy. However, in reality, the WSN is made up of heterogeneous sensors having different features and various modalities. Taking into account such diversity can improve the network performance and specifically the network lifetime.

One of the most trivial solutions in the case of heterogeneous WSN, is to partition the network into clusters and to elect energy powerful nodes as cluster heads. In each cluster, energy weak nodes send their messages to the cluster head which in turn send the fused messages to the base station directly or by using other cluster heads as intermediate nodes [4, 5]. Such solutions suite deterministic WSN deployment cases, where cluster heads can be placed at the range of each other. However, deterministic deployment is not always feasible. For example, in the case of natural disasters, or hostile supervision, the supervised area cannot be accessed directly. Sensors are scattered randomly and they should form autonomously their own network without any humain intervention. The cluster heads can be placed so far away from each other making the direct communication between them a real challenge.

In this paper, we present CLEVER (Cluster-based Energy-aware Virtual Ring Routing), a novel cluster-based routing protocol that condensates routing traffic in powerful sensors in randomly deployed WSN. For each node, CLEVER assigns the transmitting power proportional to its amount of energy. A cluster is defined as a set of energy-powerful nodes covered by each other. These powerful clusters cannot cover all the supervised area without the collaboration of other energy weak nodes. A new routing strategy is then proposed in order to take at most advantage from the strong clusters with a careful use of weak sensors when required. To optimize more efficiently the sensors energy, CLEVER adjusts nodes transmission power so that there is no transmitting energy wastage.

CLEVER uses virtual identity-based routing for intra and inter-cluster communications. Each sensor has a location-independent identifier. Virtual ring (DHT-like) routing is applied in each cluster and for global routing. CLEVER accelerates the routing process, by performing big virtual hops among energy-powerful nodes. Note that CLEVER fit well with the WSN integration in the Internet of Things [18], since nodes and data are reachable through location-independent identifiers. To the best of our knowledge, this is the first clustering routing protocol that addresses randomly deployed heterogeneous WSN, using location-independent identifiers.

Simulations results show that our approach minimizes significantly the use of energy-critical nodes which extends their lifetime and optimizes drastically sensors energy in comparison with well-known energy-aware cluster-based protocols.

This paper is structured as follows. Section 2 presents our research background. The related work is given in Section 3. Section 4 presents the problem description as well as our assumptions. Section 5 describes our proposed routing strategy CLEVER. Section 6 illustrates the simulation results. Finally Section 7 concludes this paper.

2 Background

Our proposed approach is based on the Virtual Ring Routing (VRR) protocol, which relies on Distributed Hash Tables (DHT) principles. In this section, we present the details of the DHT and VRR systems.

2.1 Distributed Hash Tables

Distributed Hash Tables (DHTs), such as Chord [20] and Pastry [21], provide a distributed lookup service similar to a hash table. Each node in a DHT network has a unique identifier (ID) and is responsible for some (key, value) pairs which have closest keys to its ID. A key is an identifier of its associated value. Depending on the application built over DHT, the value can be a data, a metadata, etc. Generally, nodes are organized virtually according to the growing order of nodes identifiers on a DHT ring (Fig. 1). The lookup function of a key returns the node responsible for that key. The associated value is then stored/retrieved in/from that node, depending on the DHT API function (PUT/GET).

A DHT system handles the nodes joining, departure, and failure. When a new node joins the network, it simply needs to know an existing node in the DHT system. It sends to this existing node a join message that will be forwarded using the routing strategy explained above until reaching the node responsible of that identifier. In this way, the new joining node is well established in the DHT system. Then, it retrieves all the (key, value) pairs that is responsible of from the old node that held them [4].

2.2 Virtual Ring Routing

Virtual Ring Routing (VRR) [9] is a DHT-based routing protocol in Ad-hoc networks. Each node in VRR has a location independent identifier. All nodes are organized into a virtual ring in an increasing order of their identifiers. Figure 2 depicts the difference between physical and virtual VRR topology.

Each VRR node maintains in its routing table:

- The next physical hops towards the virtual neighbors of the current node.
- The next physical hops towards the virtual paths that the current node participated in their setup.
- A physical neighbor set containing the identifiers of the physical neighbors.

To forward messages from a node to another, the VRR node picks from its routing table, the node having the closest

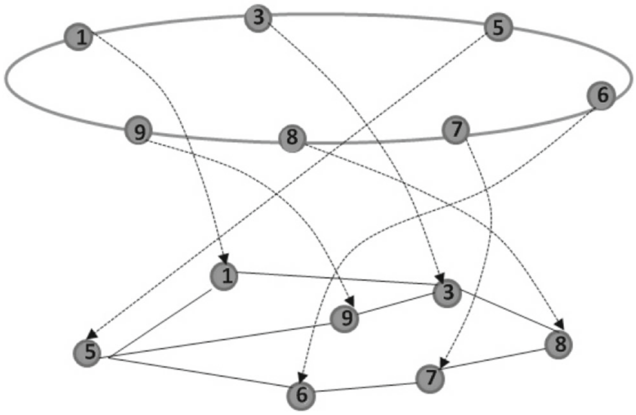


Fig. 1 Overlay and underlay DHT topology

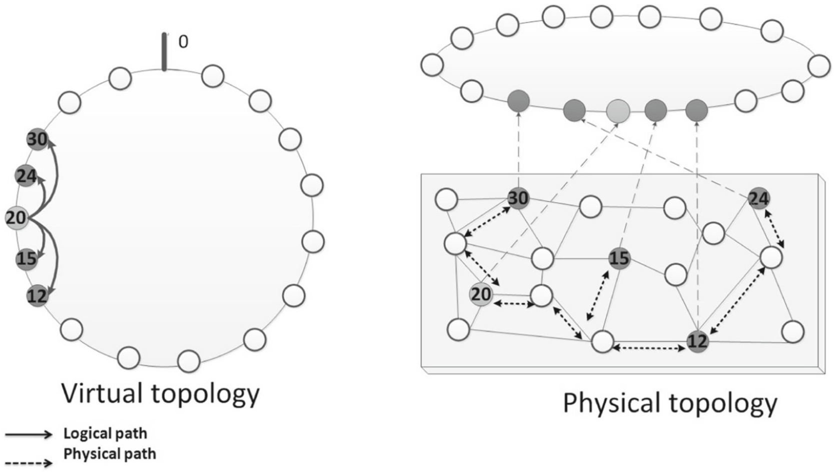


Fig. 2 Difference between virtual and physical topology in VRR

identifier to the destination, and forwards the message to its corresponding next physical hop. The nodes along a routing path store the next hop towards each path endpoint in a routing table.

In order to further improve the routing performance, VRR allows greedy hops. This means that when a physical path going between two given nodes crosses another physical path that can shorten the path to the destination, VRR switches to it.

VRR offers a suitable solution for large WSN. It offers a flexible and scalable routing process with fault tolerance. The limit of VRR is that it does not take into account nodes characteristics and especially nodes energy levels in routing. Routing without worrying about the energy level of the different nodes leads to the quick energy depletion of the energy critical nodes. This has a negative impact on the network lifetime.

3 Related work

Since CLEVER is a DHT-based routing protocol, this section compares the proposal with cluster-based and DHT-based routing protocols in WSN [8].

3.1 Cluster-based routing protocols

Network clustering is a well known solution that aims to optimize the energy consumption and extend the lifetime of the WSN.

Low-Energy Adaptive Clustering Hierarchy (LEACH) [4] is one of the first and most known cluster-based routing protocols. In LEACH, the network is partitioned into clusters. Sensors belonging to each cluster send their data to their corresponding cluster head. This latter performs data fusion and aggregation and send the resulted data to the base station.

In [5], two improvements of LEACH, Multi-hop LEACH and Energy-LEACH have been proposed. The main contribution of Multi-hop LEACH is that it allows the use of intermediate cluster heads in inter-cluster routing instead of direct communication between each cluster head and the base station. Energy- LEACH chooses in each round, nodes having greater amounts of energy.

In [6], cluster heads are chosen randomly from the set of energy powerful nodes. If the number of cluster heads chosen from this set is below the required number, other temporary cluster heads are chosen. The inter-cluster communication involves only cluster heads if their number is sufficient, and uses some temporary cluster heads otherwise. Intra-cluster routing is ensured on one hop communication between a given node and a cluster head.

Protocols proposed in [5, 6] are location based. Since our proposed contribution is DHT- based, it does not need any location information.

In [7], clusters are made up of a cluster head and its one hop neighbors nodes. In inter-cluster routing phase, when a cluster head is unable to reach another cluster head directly, it uses its nearest neighbors to forward the message to the other cluster head. The limit in this approach is that the size of these clusters is very small (one hop). This leads to the use of multiple nodes in routing which is not beneficial for the network lifetime.

3.2 DHT-based routing protocols

VRR [9], ScatterPastry [10], ScatterDHT [11] and Scalable Source Routing (SSR) [12] use the concept of DHT systems without worrying about the physical location of nodes.

Two virtual neighbors can be physically far away. Such techniques facilitate the network maintenance in churns but cause unnecessary additional steps in routing. The hierarchical topology of CLEVER minimizes the number of physical hops in routing.

Tiered Chord (TChord) [13] and SNBDT [14], are hierarchical DHT-based routing protocols. They assign additional routing roles to master nodes and try to avoid slave nodes. TChord and SNBDT can only be applied when master nodes have all the time access to power and are placed not far away from the other slave nodes. Such assumptions let application domains for these protocols so limited. In CLEVER, there are no conditions in node placements and all nodes have no access to power. This is a desirable characteristic in several application domains.

Coral-based VRR [15], classifies the nodes into different levels depending on their amounts of energy and applies VRR in each level. The protocol condensates the traffic in energy powerful nodes to avoid the failure of energy constrained nodes. The main limit of Coral-based VRR is that it still uses energy impoverished nodes in data packets routing: A virtual hop between two energy powerful nodes is achieved physically by the use of several energy constrained nodes. CLEVER, avoids sending messages to energy constrained nodes by forming superpeers clusters.

In [16], nodes are organized into multi-level virtual ring (MVR). In each level, there are some backbone nodes. These nodes are choosen depending on their connectivity to other nodes. Each normal sensor node should be associated to a given backbone in each level. To route a message, the level of the destination is first searched, then virtual routing is applied in order to reach this destination level, then the destination node. Using a multi-level routing offers certainly better routing performance than one level virtual routing. However, choosing the backbone node independently to the sensor nodes energy levels, would cause the quick energy depletion of energy critical backbone nodes. Superpeer nodes in CLEVER are chosen depending on their energy levels which avoids such problem.

4 Problem description and assumptions

In this section, we describe the network model then we specify the clustering problem that we aim to solve.

4.1 Network model

Given a set of sensors that are scattered randomly over a field, we assume that there are two types of nodes:

- Weak sensors (Peers): having a very critical amount of energy and limited computational capabilities.
- Strong sensors (Superpeers): having more important amount of energy and their computational capabilities are more powerful than those of weak sensors.

Weak sensors are points of homogenoeus Poisson Process with density λ_1 and strong nodes are points of homogenoeus Poisson Process with density λ_2 with $\lambda_2 = \alpha\lambda_1$ ($\alpha < 1$). We assume also that all nodes are location-unaware, that is they are not equipped with any positioning device such as GPS. Nodes are then inaccessible after deployment, their amount of energy is unreplenishable. Each sensor has the ability to tune its transmitting power.

4.2 Problem description

In multi-hop clustering approaches, powerful nodes communicate directly. However, the random deployment of sensors leads generally to inequitable distribution of powerful nodes. An example of such cases is given in Fig. 3. The nodes are partitioned into three regions. In the regions A

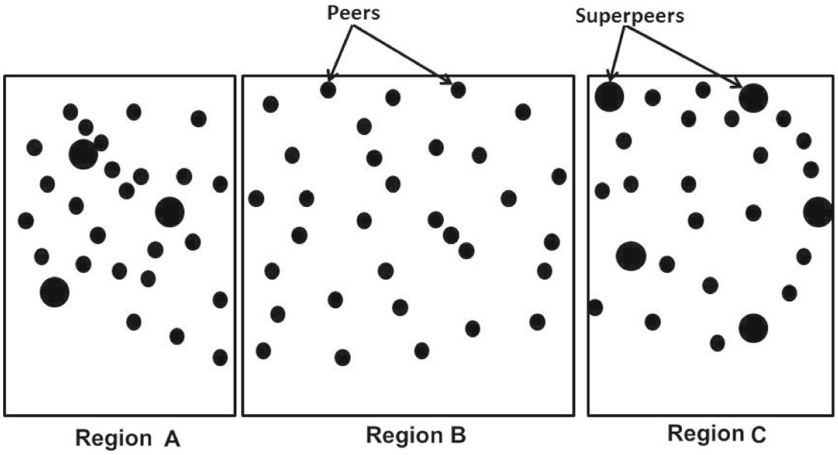


Fig. 3 Example of randomly deployed sensors

and C, there are few energy-powerful nodes surrounded by a lot of energy critical nodes. Whereas, in region B, there is a high number of weak nodes without any strong nodes. In such cases, how can we ensure the clustering process? If a weak node from the region B is elected as a cluster head, it will be more solicited in routing and this will certainly lead to its quick failure, affecting consequently the network lifetime. And if the weak nodes in region B will be associated to a cluster from region A or region C, the formed cluster will be extremely large. This solution has a negative impact on the sensors amount of energy and also causes a significant delay on messages transmission.

In this paper, we propose a clustering as well as intra and inter-cluster routing strategies that solve the mentioned problem in a totally distributed and energy-aware manner.

5 Design of CLEVER

This section describes the CLEVER system. First, we give an overview of the proposed architecture. Second, we describe the routing process. Third, we detail the node joining and departure operations.

5.1 Architecture overview

Each node has a location independent identifier. It can be the hash of the node MAC address. We assume that each node is stationary and that there are two energy levels: peers energy level and superpeers energy level. The sink node has also a location independent identifier and is considered as an energy powerfull node.

Sensors adjust their initial transmission power depending on their amount of energy. Energy powerful nodes (having important transmission power) can reach farther nodes, forming superpeers clusters (see Fig. 5). In each cluster, superpeers communicate directly with each other using the VRR routing protocol. This enables fast routing process since it is likely to encounter, through direct communication, a superpeer close *numerically* to the sink node. Since superpeers in different clusters cannot communicate directly with each other, there is a need to use some weak nodes (peers) as intermediate nodes. Peers act only as bridges between clusters. In order to ensure an efficient inter-cluster routing, all superpeers are placed into a global superpeers ring and all nodes (superpeers and peers) are placed into a global virtual ring. Unlike the majority of cluster-based routing protocols, a peer is not related to a single superpeer. It has several physical neighbors to reach the sink since it belongs to a virtual ring. *This would avoid the client/server relationship between the peer and the superpeer.*

5.2 Data packets routing

Algorithm 1 CLEVER data packets routing pseudocode

```
procedure RECEIVE($dataPkt, src, dst, Virt, Sender$)
    if dst==me then
        pktReceived=true
    else
        if status=superpeer then
            if me==Virt then
                newVirt=rtableCluster.closestVirt(me, dst)
                if newVirt!=NULL then
                    newPhy=rtableCluster.nextPhys(newVirt)
                else
                    newVirt=rtableSuper.closestVirt(me, dst)
                    newPhy=rtableSuper.nextPhys(newVirt)
                send(dataPkt,me,dst,newVirt,sender) to newPhy
            else
                newVirt=rtableCluster.closestVirt(me, dst)
                if $|newVirt - dst| < |Virt - dst|$ then
                    Virt=newVirt
                newPhy=rtableCluster.nextPhys(Virt)
                if newPhy==NULL then
                    newPhy=rtableSuper.nextPhys(Virt)
        else
            newPhy=rtablePeer.nextPhys(Virt)
            send(dataPkt,me,dst,newVirt,sender) to newPhy
end procedure
```

Each peer has a routing table. The routing table maintains the physical neighbors set of the current node. In each routing table entry, the identifiers of the two extremities of virtual paths that the current node participated in their construction, as well as the two next physical hops that lead to each extremity are maintained.

Each superpeer has two additional routing tables: Cluster routing table and global superpeers routing table. The structure of these routing tables is the same as the peers routing table. The only difference is that the virtual paths that are maintained in the cluster routing table are related to the cluster only and the physical neighbors are only superpeers. Similarly, superpeers routing table maintain virtual paths between the virtual neighbor superpeers. These two routing tables are used for inter and intra cluster communications at superpeer level.

Data packets are routed mainly by superpeers. In each cluster, only superpeers are involved in the routing process since each superpeer can communicate directly with at least another superpeer of its cluster. When a cluster receives a

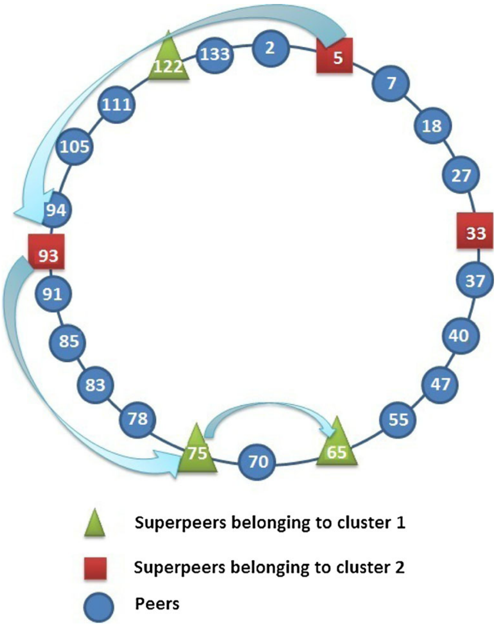

5
7
18
27
33
37
40
47
55
65
70
75
78
83
85
91
93
94
105
111
122
133
2
Superpeers belonging to cluster 1
Superpeers belonging to cluster 2
Peers

Fig. 4 CLEVER virtual routing

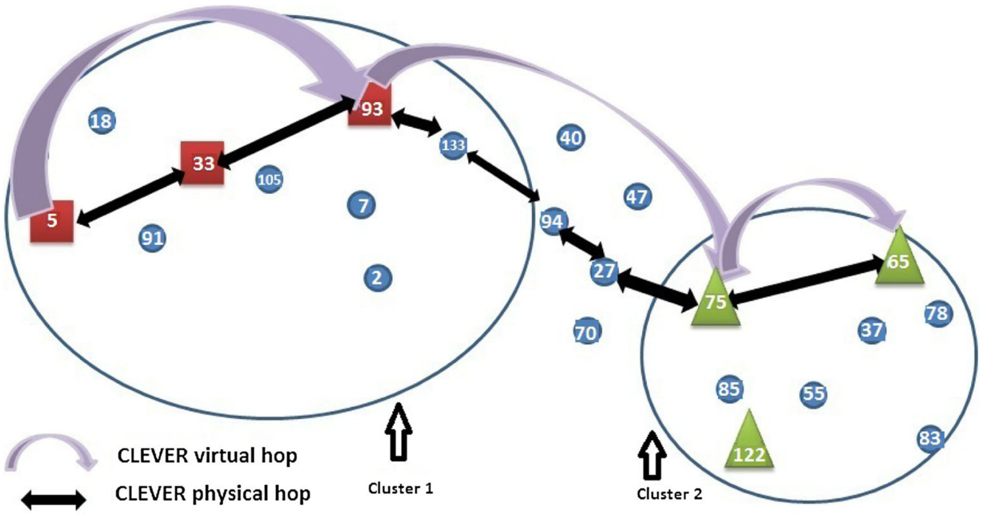
data packet, VRR is used to forward the packet to the closest superpeer to the destination in this cluster. From that node, intermediate peers forward the message to the next superpeer in the global ring. This procedure is repeated until reaching the destination. The pseudo-code of the routing process is depicted in algorithm 1. In this algorithm, a data packet *datapkt* whose destination is *dst* and source is *src* is received by the current node *me* from the previous physical hop *sender*. This current node picks from its routing table the next virtual hop *newVirt* and the next physical hop *newPhy* to reach *newvirt*. *closestVirt* and *nextPhys* are the methods that pick from the corresponding routing table, the next virtual hop and the next physical hop. If the curent node is a superpeer, the routing table that should be picked is *rtableCluster*: the cluster routing table. If the current node is a superpeer having the closest identifier to the destination in its cluster, the routing table that should be picked is *rtableSuper*: the superpeers routing table. If the current node is a peer node, the routing table that should be picked is *rtablePeer*: the peers routing table.

Figures 4 and 5 present an example of respectively virtual and physical routing in CLEVER between nodes 5 and 65. For the sake of simplicity, we do not use greedy hops in VRR and CLEVER in the figures. As depicted in Fig. 4, CLEVER achieves very large hops towards the superpeers. Physically, the VRR is applied in the first cluster (made of superpeers 5, 33, 93) (Fig. 5), in such a way that the node 93 is reached by using only superpeers nodes. Node 93 picks from its global superpeers routing table the superpeer having the closest identifier to the destination: the node 75. To reach this node, intermediate peers are used in the physical routing. Once a superpeer 75 is reached, the VRR routing is applied in its corresponding cluster.

5.3 Node joining

The node joining process depends on the type of sensor. The first step is to specify this type as well as the initial transmission power according to the amount of energy. Then, the node should be well positioned in the global peers ring: in this ring, all nodes in the network independently from their natures are organized in an increasing order of their identifiers. As in VRR, the new joining node sends a setup request message to one of its physical neighbors considered as proxy, with its own identifier. This physical neighbor

Fig. 5 CLEVER physical routing



picks from its routing table the next virtual hop and forwards this setup request message to its corresponding next physical hop. This procedure is repeated until reaching the node having the closest identifier to the joining node. The destination node sends its virtual neighbors set to the new joining node, which can then construct its virtual neighbors. If the joining node is a superpeer, it should be positioned in the global peers ring, the cluster ring and global superpeers ring.

Algorithm 2 CLEVER joining process algorithm

```
if $energy \geq threshold1$ then
    Status=superpeer
    Transmit_power=Psuperpeer
else
    Status=peer
    Transmit_power=Ppeer
VRR_joining_process (0, node id)
if Status==superpeer then
    if $nbReceivedHelloCluster \geq 2$ then
        for $i = 1$ to $nbReceivedHello$ do
            VRR_joining_process (cluster_id,node id)
        end for
        for $i = 1$ to $nbMyVirtNC2$ do
            Send_C_Merge (me,MyVirtNC2[i],MyVirtC1,nbMyVirtNC1)
        end for
        for $j = 1$ to $nbMyVirtNC1$ do
            Send_C_Merge (me,MyVirtNC1[j],MyVirtNC2,nbMyVirtNC2)
        end for
    else
        if nbReceivedHelloCluster == 1 then
            VRR_joining_process (cluster_id, node id)
        else
            Newcluster (node id)
    if stabilized then
        newPhy=rtablePeer.nextPhys(MyImmediate_succ)
        send_setupRSuperpeer(me,MyImmediate_succ) to newPhy
procedure RECEIVE_C_MERGE($src, dst, VirtN, nbVirtN$)
    if me=dst then
        for $i = 1$ to $nbVirtN$ do
            for $j = 1$ to $nbVirtN$ do
                if IsCloser(Virt_N_set[i], MyVirtN[j]) then
                    send_setupR_C(me,Virt_N_set[i], proxy,src )
                    send_teardownC (me,MyVirtN[j],me,nh)
            end for
        end for
    else
        nh=nextHop(src, dst)
        Send_C_Merge (src,dst,VirtN,nbVirtN)
end procedure
```

If the joining node is a superpeer, it checks if it belongs to an existing cluster or not. This check is ensured by HELLO message exchanging. HELLO messages that are sent by superpeers contain a field indicating the cluster identifier to which it belongs. These HELLO messages are transmitted with the maximum transmission power of the source node in order to discover the maximum of physical neighbors. When a joining superpeer receives a HELLO message from an existing superpeer, it perceives that it belongs to an existing cluster. If this is the case, it initiates a local joining procedure.

If the joining superpeer does not belong to any existing cluster, it forms a new cluster that contains only this joining superpeer. The cluster will have as an identifier, the identifier of this new joining node.

A joining superpeer can merge multiple disjoined clusters. It can detect this, when it receives HELLO messages having different cluster identifiers. We call this superpeer a guard node. As it is a common node in the different clusters, the nodes can exchange the nodes identifiers enabling them to merge into a common ring. To realize this, the guard node searches its place in each virtual ring of the attached clusters and merges the different clusters.

When all the nodes are stabilized, superpeers start the global superpeers ring joining process. All the joining procedure is depicted in algorithm 2. Table 1 depicts the algorithm2 variables, functions and their significations.

5.4 Node leaving

When a node fails, it leaves the rings to which it belongs. When a superpeer becomes a peer, it leaves the superpeers rings (cluster and global superpeers rings). These departures are detected by physical neighbors.

Superpeer failures or transformation into peers may have two possible circumstances:

1. All superpeers belonging to the same cluster of the failed superpeer are still reachable through intermediate superpeers as depicted in Fig. 6. We call this failure Local Failure (LF).
2. The failed superpeer was the only superpeer that connects the different sides of the cluster (see Fig. 7a). In this case, some superpeers that belonged to the same cluster of the failed superpeer become unreachable. This means that superpeers in the same cluster become unable to deliver messages to these superpeers without the need of intermediate peers. The cluster of the failed superpeer is then split into different clusters as depicted in Fig. 7b. We call this failure Splitting Cluster Failure (SCF).

5.5 Failure repair

5.5.1 Peer failure repair

The peer failure repair is similar as in VRR. CLEVER tries to repair the failure locally. This means that if a node fails, the routing protocol tries to replace the failed node by

Table 1 Functions/Variables and their significations

Variable/Function	Signification
threshold	energy level of superpeers
Ppeer	transmission power of peers
Psuperpeer	transmission power of superpeers
nodeid	node identifier of joining node
cluster_id	cluster identifier
VRR_joining_process	applies VRR joining in the cluster identified by cluster_id
nbReceivedHelloCluster	number of received Hello messages from superpeers belonging to different clusters
Send_C_Merge	merges all the clusters having the new superpeer as a common node
nbMyVirtNC	the number of virtual cluster neighbors of the current node
Newcluster	creates a new cluster that contains the new joining superpeer
stabilized	tests if the node is well positioned into the global virtual ring and in its cluster ring if it is a superpeer
rtablePeer	the global peers routing table
setupRSuperpeer	setup request superpeer message sent by a joining superpeer to know its position in the superpeers ring
SetupSuperpeer	sent as a response to the setupRSuperpeer message from the virtual neighbor superpeer to the new superpeer
setupRC	setup request superpeer cluster message sent by a new joining superpeer in order to know its virtual position in the cluster ring
setupC	sent as a response to the setupRC message from the virtual neighbor in the cluster ring
teardownC	a message that deletes a virtual path no more available in a cluster
IsCloser	tests if the virtual neighbor contained in C_Merge message is closer than the virtual neighbor(*MyVirtN)* of the current node
vsetClusterAdd	adds the new virtual neighbor to the cluster virtual neighbors set
add_EntryCluster	adds a new entry in the cluster routing table containing the path to the new virtual neighbor
remove(MyVirtN, nodeA)	removes the node A from the virtual neighbors set
me	the identifier of the current node
src	the message source node
dst	the message destination node
sender	the message previous physical hop
proxy	the first node that forwards the setupRC message
pid	the path identifier

another node without the need to reconstruct the entire path. If the endpoint is a physical neighbor to the current node, this latter patches directly to this endpoint. If the node A, the next hop after the failed node, is a physical neighbor to the current node, the failed node can be bypassed and A will be directly the next hop. Otherwise, the protocol tries to find another physical neighbor B that has a link to the node A. If such a neighbor exists, the failed node will be replaced by a link from the current node to the node B and from the node B to the node A.

If this local repair fails, all the paths that traverse the failed peer are teardown and reconstructed. The signaling messages are routed using CLEVER routing protocol.

5.5.2 Superpeer failure repair

In the case of a superpeer failure, the global peers ring, the superpeers cluster ring, and the global superpeers ring should be updated. If a superpeer is transformed into a peer, only superpeers cluster ring, and the global superpeers ring should be updated. CLEVER tries initially to repair locally the paths that traversed the failed superpeer.

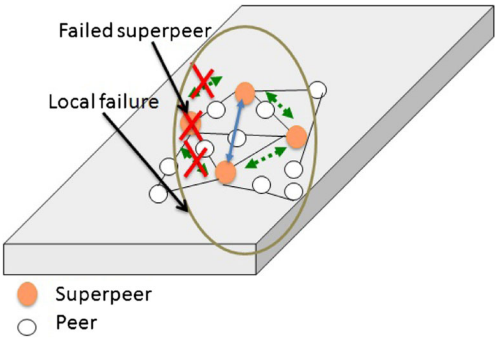
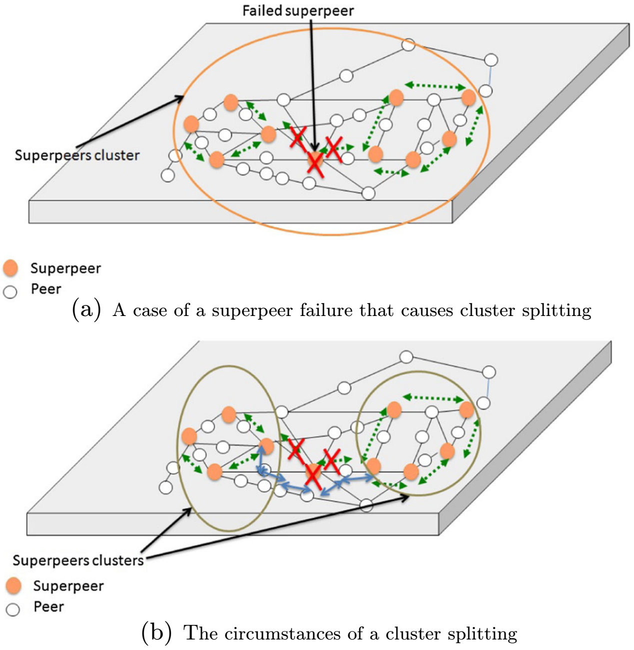


Fig. 6 Superpeer local failure



(a) A case of a superpeer failure that causes cluster splitting

(b) The circumstances of a cluster splitting

Fig. 7 Causes and circumstances of a cluster splitting

Most of local failure cases of superpeers cluster ring are repaired locally. If this local repair fails, CLEVER sends teardownCluster messages to delete all the paths in the cluster that traverse the failed superpeer and reconstructs new ones. Each superpeer that receives the teardownCluster message deletes from its cluster table any entry that contains the failed superpeer as next hop. CLEVER reconstructs new virtual cluster path by sending setup_requestCluster message.

When a superpeer receives a teardownCluster message for a virtual path and it is one of the two endpoints of this message, it removes the other endpoint from its virtual cluster neighbors set and sends a setup_requestCluster message to the removed endpoint in order to reconstruct the path. If this message is retransmitted up to five times without any response, the node is considered as dead or unreachable.

In such a case, the node sends the setup_requestCluster message to the superpeer having the closest identifier to the dead virtual superpeer in the same cluster. If there is no response, CLEVER realizes that the old cluster is split after the failure of this superpeer. Hence, the node that wants to reconstruct a virtual cluster path reinitializes its virtual superpeer set in the old cluster. It restarts a setup phase in order to know the new virtual superpeer neighbors in the new cluster with the new virtual paths.

5.6 Optimal transmitting power for sensors

In this section, we specify the optimal transmitting power for strong sensors in order to keep the network structure, and for all sensors in order to ensure k-average degree. A complete study can be found in our previous work in [19].

5.6.1 Optimal transmitting power for strong sensors

It is trivial that the more the transmitting power of the strong nodes increases, the more the number of contacted weak sensors decreases. However, increasing indefinitely the transmitting power will surely deteriorate the energy of the powerful nodes and will cause major changes in the network: the number of strong nodes decreases, since they become weak. In order to keep the network structure, the transmitting power of these strong nodes should be tuned in a way that ensures that during the lifetime of the weak nodes, strong nodes remain strong and do not switch to the weak state (i.e. the amount of energy of strong nodes should not drop below the weak nodes threshold).

Assuming that the relation between the transmitting powers of strong nodes and weak nodes is given by:

$$Pt_2 = \beta.Pt_1 \tag{1}$$

with $\beta > 1$

We aim to calculate the optimal value of β keeping the network structure.

Assume that a weak sensor is able to transmit N messages before its death and that the energy of a sensor is mainly spent by transmitting messages (we ignore the amount of energy spent in processing, receiving and idle states). This can be expressed as follows:

$$E_{Weak} \approx N.(T_{transmission}.Pt_1) \tag{2}$$

where $T_{transmission}$ is the time spent in transmitting and receiving a message.

The amount of energy of strong nodes should not drop below the weak nodes threshold. This condition is expressed as follows:

$$E_{Strong} - E_{Weak} \approx N.\beta.(T_{transmission}.Pt_1) \tag{3}$$

From Eqs. 2 and 3 we obtain:

$$\beta = (E_{Strong} - E_{Weak})/E_{Weak} \tag{4}$$

Note that β should be bigger than 1, that is E_{Strong} should be bigger than $2 * E_{Weak}$. The maximum transmitting power of the strong nodes is:

$$Pt_2 = (E_{Strong} - E_{Weak}).Pt_1/E_{Weak} \tag{5}$$

5.6.2 Optimal transmitting power for a k-average degree

Each sensor in a network should be connected to an optimal number of neighbors called *magic number* in order to ensure efficiently its functionnalities [22, 26]. For example in [23], there is an optimal neighbors number that ensures optimal throughput.

In this section, we estimate the nodes transmitting powers in a way that each node has in average k physical neighbors i.e. it has k-average degree.

In order to ensure this k-average degree, the sensors transmitting power should use a minimal transmitting power Pt_{min}. When the transmitting power P_t is greater than Pt_{min}, the number of the neighbors N(P) is greater than k. Hence we have [24] :

$$P(Pt_{min} <= P_t) = P[N(p) >= k] = 1 - P_k \tag{6}$$

At the case of heterogeneous WSN having two types of sensors (weak and strong sensors), the probability of k-coverage is [25]:

$$P_{cover-k} = 1 - \sum_{m=0}^{k-1} \left[\sum_{j=0}^{m} P_1(j).P_2(m-j) \right] \tag{7}$$

$P_1(j)$ is the probability that there are j nodes in the area S_1

$$P_1(j) = \frac{(S_1.\lambda_1)^j}{j!}.e^{-\lambda_1.S_1} \tag{8}$$

$P_2(m-j)$ is the probability that there are m-j nodes in the area S_2

$$P_2(m-j) = \frac{(S_2.\lambda_2)^{m-j}}{(m-j)!}.e^{-\lambda_2.S_2} \tag{9}$$

With S_1 and S_2 are respectively the surface area covered by a weak and strong sensors.

$$S_1 = \pi.r_1^2 \tag{10}$$

$$S_2 = \pi.{r_2}^2 = \pi.\beta.r_1^2. \tag{11}$$

The transmitting ranges of weak and strong nodes are respectively:

$$r_1 = C.\sqrt{Pt_1} \tag{12}$$

$$r_2 = \sqrt{\beta}.r_1. \tag{13}$$

$$C = \sqrt{\frac{wavelength^2 \cdot Gt \cdot Gr}{RxThresh \cdot 16 \cdot \pi^2 \cdot L}} \tag{14}$$

RxThresh is the receiving threshold in the network interface, Gt and Gr are respectively the transmit and reception gain, L is the system Loss.

Equation 7 is related to coverage problem. In order to study the k-average degree problem, we should treat the case for strong nodes and the case for weak nodes separately.

If the current node is a strong node it can communicate with strong nodes that are within its communication range and it can also communicate with weak nodes if the current strong node is within their communication range. The probability that the average degree of a strong node is k:

$$P_k(strong) = \left(1 - \sum_{m=0}^{k-1} \left[\sum_{j=0}^{m} P_1(j).P_2(m-j) \right] \right).P(strong) \tag{15}$$

P(strong) is the probability that the current node is a strong node and is equal to $\frac{\alpha}{1+\alpha}$.

If the node is a weak node, it can only communicate with nodes in its communication range independently from their natures.

$$P_k(weak) = \left(1 - \sum_{m=0}^{k-1} \left[\sum_{j=0}^{m} P_1(j).P_3(m-j) \right] \right).P(weak) \tag{16}$$

$P_3(m-j)$ is the probability that there are m-j strong nodes in the area S_1 and P(weak) is the probability that the current node is a weak node and is equal to $\frac{1}{1+\alpha}$.

$$P_3(m-j) = \frac{(S_1.\lambda_2)^{m-j}}{(m-j)!}.e^{-\lambda_2.S_1} \tag{17}$$

Then, the probability that a given node has a k-average degree:

$$\begin{aligned} P_k &= P_k(strong) + P_k(weak) \\ &= 1 - \sum_{m=0}^{k-1} \left[\sum_{j=0}^{m} P_1(j)P_2(m-j)P(strong) \right. \\ &\quad \left. + P_1(j)P_3(m-j)P(weak) \right] \end{aligned} \tag{18}$$

From Eqs. 15 and 16, we obtain:

$$\begin{aligned} E[Pt_1] = \int_0^{\infty} \sum_{m=0}^{k-1} \left(\sum_{j=0}^{m} (P_1(j).P_2(m-j).P(strong) \right. \\ \left. + P_1(j).P_3(m-j).P(weak)) \right) \mathrm{d}Pt_1 \end{aligned} \tag{19}$$

$$E[Pt_1] = A + B \tag{20}$$

Where:

$$A = \int_0^{\infty} \sum_{m=0}^{k-1} \left[\sum_{j=0}^{m} P_1(j).P_2(m-j).P(strong) \right] \mathrm{d}Pt_1 \tag{21}$$

$$B = \int_0^\infty \sum_{m=0}^{k-1} \left[\sum_{j=0}^{m} P_1(j).P_3(m-j).P(weak) \right] \mathrm{d}Pt_1 \tag{22}$$

$$A = \sum_{m=0}^{k-1}\sum_{j=0}^{m} \frac{\lambda_1.\pi.C^2.(\alpha.\beta)^{m-j}.P(strong)}{j!.(m-j)!}. \int_0^\infty {Pt_1}^m.e^{-\lambda_1.\pi.C^2.Pt_1.(1+\alpha.\beta)}\,\mathrm{d}Pt_1. \tag{23}$$

Using the integration by substitution, we obtain:

$$A = \sum_{m=0}^{k-1}\sum_{j=0}^{m} \frac{\Gamma(m+1).(\alpha.\beta)^{m-j}P(strong)}{j!(m-j)!\lambda_1.\pi.C^2.(1+\alpha.\beta)^{m+1}}. \tag{24}$$

where Γ function: $\Gamma(b) = \int_0^\infty a^{b-1}e^{-a}\mathrm{d}a$

$$B = \sum_{m=0}^{k-1}\sum_{j=0}^{m} \frac{{\lambda_1}^m.\pi^m.C^{2m}.(\alpha)^m - j.P(weak)}{j!.(m-j)!}. \int_0^\infty {Pt_1}^m.e^{-\lambda_1.\pi.C^2.Pt_1.(1+\alpha)}\,\mathrm{d}Pt_1. \tag{25}$$

Using the integration by substitution, we obtain:

$$B = \sum_{m=0}^{k-1}\sum_{j=0}^{m} \frac{\Gamma(m+1).(\alpha)^{m-j}P(weak)}{j!(m-j)!\lambda_1.\pi.C^2.(1+\alpha)^{m+1}}. \tag{26}$$

Hence, the average transmitting power Pt_1 that should be used by a weak node to has k-average degree is

$$E[Pt_1]=\sum_{m=0}^{k-1}\sum_{j=0}^{m}\frac{m!\left(\alpha^{m-j+1}.\beta^{m-j}.(1+\alpha)^{m+1}+\alpha^{m-j}.(1+\alpha.\beta)^{m+1}\right)}{j!(m-j)!\lambda_1.\pi.C^2.(1+\alpha.\beta)^{m+1}.(1+\alpha)^{m+2}} \tag{27}$$

The average transmitting power Pt_2 that should be employed by a strong node is then:

$$E[Pt_2] = \beta.E[Pt_1] \tag{28}$$

6 Simulation results

6.1 Experimental setup

All the simulations were carried out using NS2 [17]. In all simulations, each sensor node sends a packet to the sink in each round. The round duration is 20 seconds. The network size is 100 static nodes distributed over 185 m * 185 m plane. We vary the number of nodes from 50 to 200. In all simulations, each sensor node sends a packet to the sink in each round. The round duration is 20 seconds. There are two types of nodes: Superpeers and peers having as initial amount of energy 10 joules and 2 Joules respectively. The transmission range of peers is 20 m and the transmission range of superpeers is 40 m. All sensors are placed randomly in order to study the case of random network deployment.

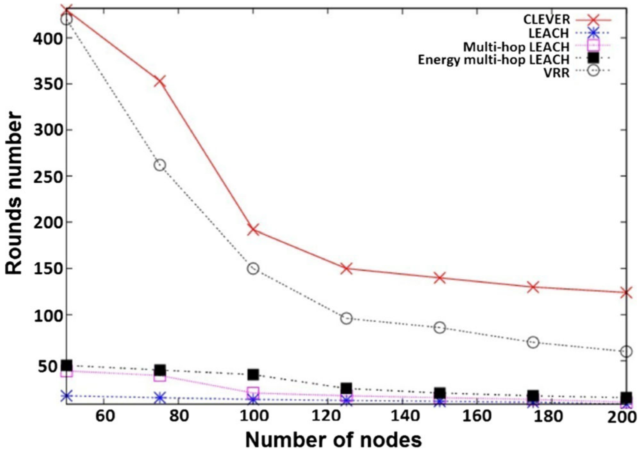


Fig. 8 Network lifetime

In the first simulation set, we fixed the percentage of superpeer nodes to 10 % of the total number of nodes in the network.

We compared the performance of CLEVER, VRR, LEACH, Multi-hop LEACH and energy-Multi-hop LEACH. The latter is the combination between energy-LEACH and multi-hop LEACH proposed in [5]. We evaluate the consumed energy, the end to end delay and the physical path length.

In the second simulation set, we vary the percentage of superpeers from 5 % to 30 % and study its impact on sensors lifetime, hops number, consumed energy and end-to-end delay.

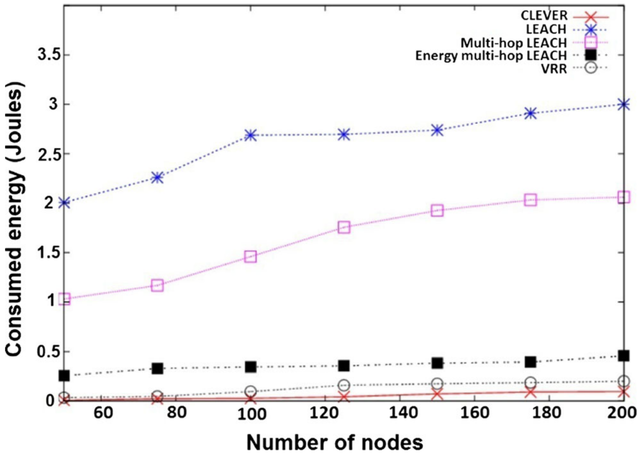


Fig. 9 Mean consumed energy with respect to the nodes number

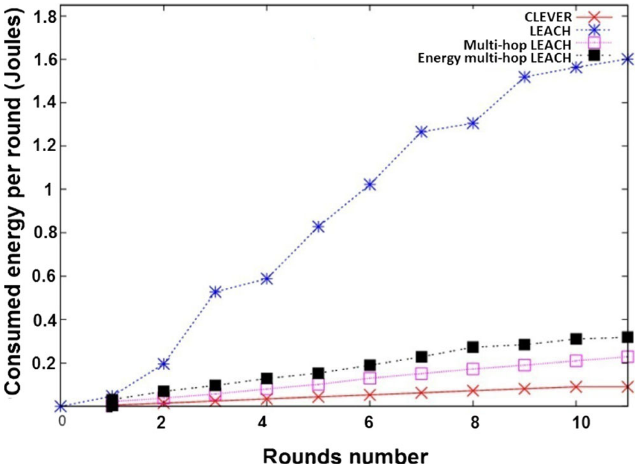


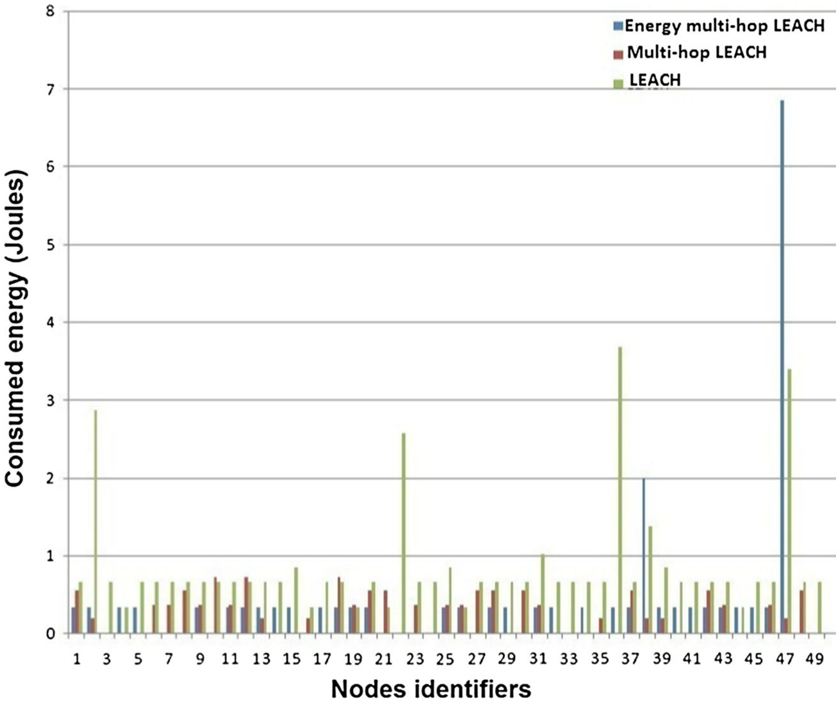
Fig. 10 Mean consumed energy with respect to the rounds number in a 175-nodes network

6.2 Performance evaluation

Figures 8 and 9 present respectively the network lifetime and the mean consumed energy with respect to the nodes number. Figure 8 shows clearly that the first node in CLEVER dies later than in the other routing protocols. This has multiple reasons. First, CLEVER uses a clustering process that avoids the use of energy critical nodes which extends their lifetime. Whereas, VRR does not use any clustering process and does not take into account nodes residual amount of energy. LEACH divides the network into clusters but the cluster heads are chosen randomly which deteriorates their amount of energy. Also LEACH uses direct communication between the cluster heads and the base station. This direct communication requires high transmitting power which leads to the quick failure of nodes. In spite of the fact that energy Multi-hop LEACH and Multi-hop LEACH use sensors having location information while the nodes in CLEVER are location unaware, the nodes in CLEVER die later. This is due to the inter-cluster routing used by energy Multi-hop LEACH and multi-hop LEACH. Effectively, when a base station is faraway from a given cluster head, these latters, use other intermediate cluster heads to reach this base station, independently from the distance between the cluster heads. However, these cluster heads can be placed so faraway from each other, and communicating directly with them leads to a lot of energy wastage. When a cluster head is placed faraway from another cluster head, CLEVER does not ensure direct communication with them, and favors the use of intermediate weak nodes. Such strategy is more beneficial to the cluster heads and extends their lifetime.

Figure 9 shows that CLEVER consumes globally less energy than the other compared protocols in all network sizes. Effectively, the consumed amount of energy in 200-nodes network using CLEVER, represents only 3.16 %, 4.604 %, 20.78 % and 47.43 % of respectively the amounts of energy consumed by LEACH, Multi-hop LEACH, Energy Multi-hop LEACH and VRR. The energy

Fig. 11 Consumed energy per node in LEACH, Multi-hop LEACH and Energy multi-hop LEACH



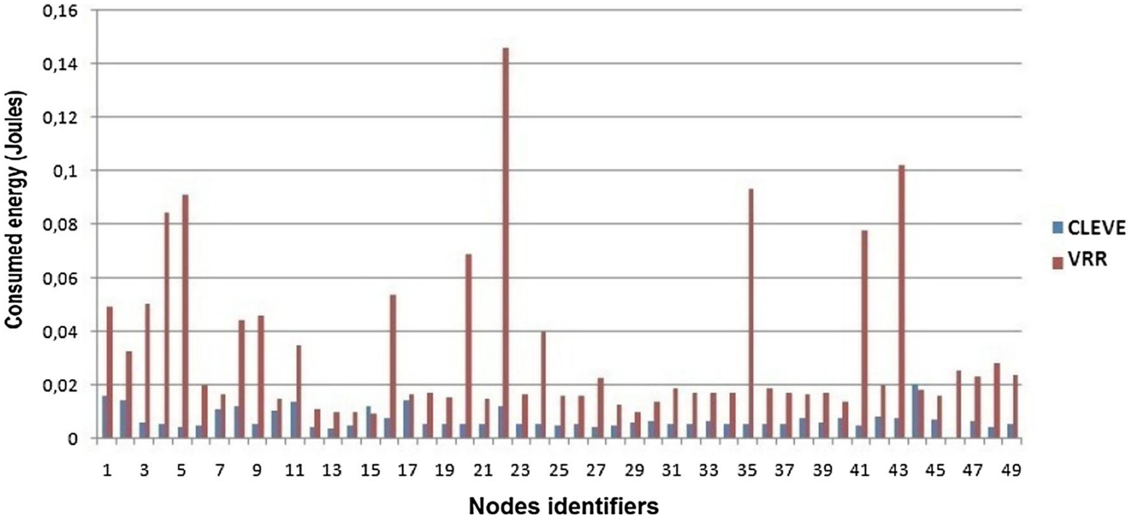


Fig. 12 Consumed energy per node in CLEVER and VRR

preservation in CLEVER is due to the superpeer nodes that are more solicited in routing than the other nodes. These superpeers have more important transmission range enabling them to reach the destination after some few intermediate nodes which reduces the use of the weak sensors that are only used as bridges between cluster superpeers. VRR consumes more energy since it considers that all nodes are homogeneous and uses a flat topology which leads to longer paths to reach the final destination. These paths are generally built by weak nodes which consumes lot of energy in comparison to CLEVER.

Cluster-heads in LEACH are chosen randomly so they can be energy starving and faraway from the sink. Thus, they need to increase at most their transmitting power to reach this sink. Such increase, consumes lot of energy. Despite of the energy-aware selection of the cluster heads in Energy multi hop LEACH, this protocol is too energy consuming since it uses only the cluster heads in the inter-cluster routing even if they are far from each other.

Figure 10 measures the variation of the mean sensors remaining energy along the time in a 175-nodes network. This figure shows how the energy is optimised from a round to another using CLEVER.

The consumed amounts of energies by each node in a 50-nodes network are given in Figs. 11 and 12. These figures show that these amounts are so limited in CLEVER in comparison to the other protocols. Also, these amounts are so close to each other in CLEVER, whereas, they are extremely different from a node to another using the other protocols. This proves the equitability of our approach that does not condensate all the traffic in only some few nodes.

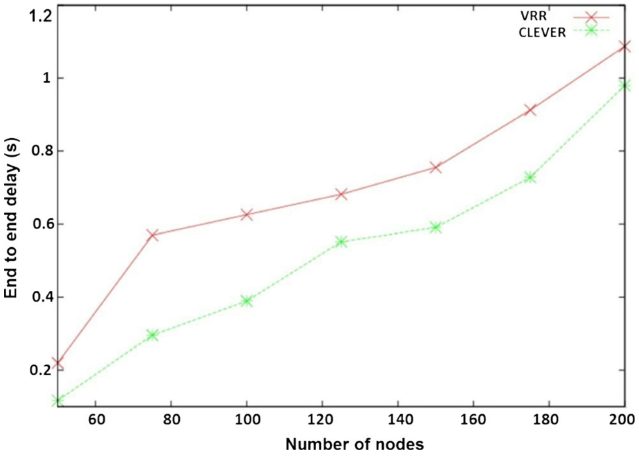


Fig. 13 Transmission delay

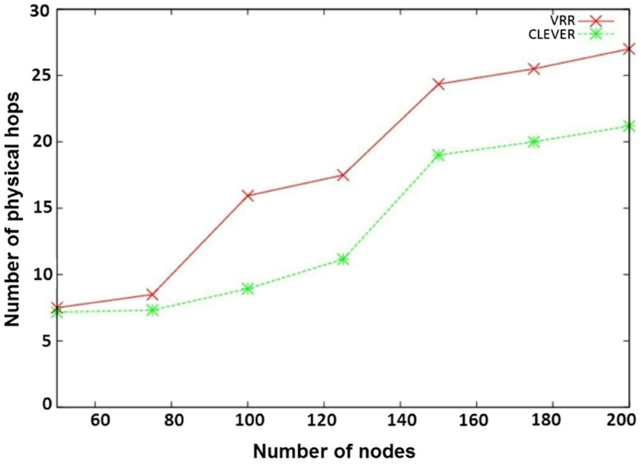


Fig. 14 Hops number

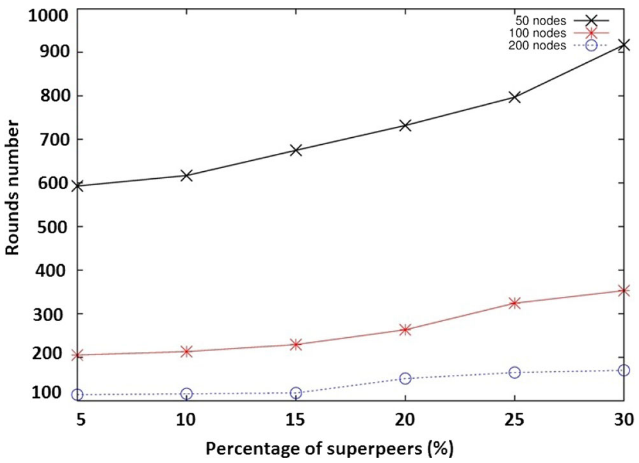


Fig. 15 Network lifetime

We compared in Figs. 13 and 14 the performance of CLEVER and VRR in terms of packets transmission delay and hops number in different network sizes. As depicted in these figures, CLEVER drastically outperforms VRR and achieves routing process in fewer hops and shorter delays. This is due to the hierarchical structure of CLEVER that allows to reach the destination in only some intermediate hops. These hops are mainly ensured by energy-powerful nodes.

Figure 15 depicts that when the percentage of superpeers increases, the lifetime of nodes increases drastically in all network sizes. This can be explained by the fact that when the number of superpeers increases, the probability of having direct communication between these superpeer increases, which leads to the fusion of multiple superpeer clusters. When the size of clusters increases, routing process is done mainly by strong nodes which lightens significantly the trafic crossing weak nodes.

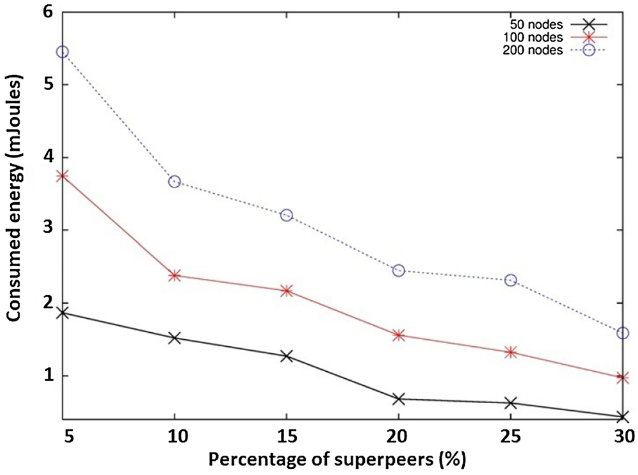


Fig. 16 Consumed energy by weak nodes per round

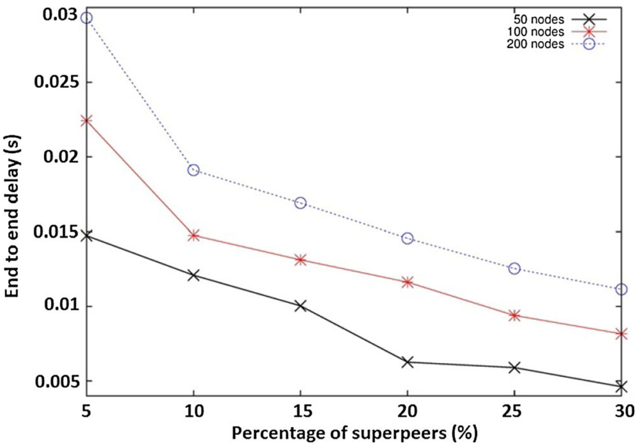


Fig. 17 Average end to end delay

Figure 16 presents the mean amount of energy consumed by weak nodes in a routing round. When the percentage of superpeers increases, the weak nodes consume less energy. Indeed, weak nodes would be rarely used in the routing process, ensured mainly by superpeer nodes.

The mean end to end delay per round is depicted in Fig. 17. This figure shows that superpeers percentage and end to end delay are inversely proportional. This is because most of routing hops are ensured by superpeers. These superpeers have more important range which allows them to reach farther nodes quickly and without the use of intermediate weak nodes.

The average hops number in different network sizes is shown in Fig. 18. This figure depicts that when the number of superpeers increases, the average number of physical hops in the routing process decreases drastically. This is also due to the extended range of superpeers that extends the cluster size and reduces the number of nodes that will act as bridges between the different clusters.

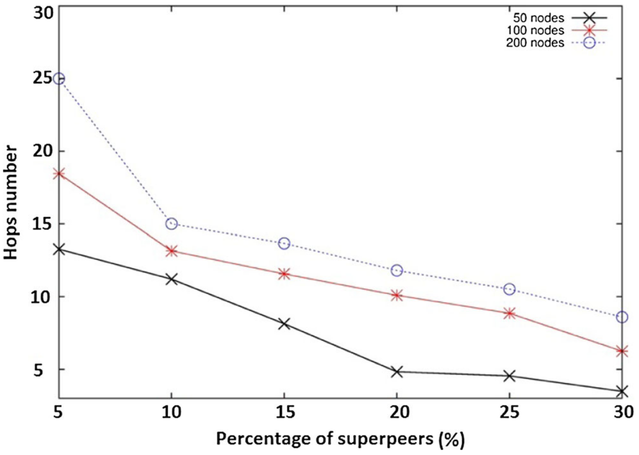


Fig. 18 Average hops number

7 Conclusion

In this paper, we have proposed CLEVER, an energy-aware identity-based routing protocol that increases the sensors lifetime in randomly deployed WSN. The transmitting powers of sensors are proportional to their amounts of energy. Consequently, the probability of having direct communication between two energy powerful nodes is increased, forming thus superpeers clusters. Intra/inter cluster communications are performed through a hierarchical virtual ring routing approach. Therefore, virtual and physical routing paths are reduced, compared to existing identity-based routing approaches.

Simulation results show that routing performance of CLEVER are much better than VRR, LEACH, Multi-hop LEACH and Energy multi-hop LEACH. These results reflect also, a good impact on the network lifetime that is clearly extended in CLEVER. Simulations have also shown that increasing the percentage of superpeers in the network extends sensors lifetime and minimizes the hops number and consumed energy. Increasing their transmitting powers has also the same impact. However, their powers should be well adjusted in order to keep their states for a long time.

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2008) Wireless sensor networks: A survey. Computer Networks 38:393–422
2. Yick J, Mukherjee B, Ghosal D (2008) Wireless sensor network survey. Comput Netw 52:2292–2330
3. Isabel D, Falko D (2009) On the lifetime of wireless sensor networks. ACM Trans Sens Netw 5:1–39
4. Heinzelman WR, Chandrakasan A, Balakrishnan H (2000) Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the international conference on system sciences, Hawaii, 4-7, Jan, IEEE, USA
5. Fan X, Song Y (2007) Improvement on LEACH protocol of wireless sensor network. In: Proceedings of the international conference on sensor technologies and applications, Valencia, 14-20 October, IEEE, USA, pp 260–264
6. Israr N, Awan I (2008) Coverage based inter cluster communication for load balancing in heterogeneous wireless sensor networks. J Telecommun Syst 38:121–132
7. Boukerche A, Martirosyan A (2007) An energy-aware and fault tolerant inter-cluster communication based protocol for wireless sensor networks. In: Proceedings of the IEEE global communications conference (GLOBECOM'07), 26-30 November, Washington, DC, IEEE, USA
8. Fersi G, Louati W, Ben Jemaa M (2012) Distributed Hash Table-based Routing and Data Management in Wireless Sensor Networks: a Survey. ACM/Springer Wireless Networks
9. Matthew C, Miguel C, Edmund BN, Greg O, Antony R (2006) Virtual ring routing: Network routing inspired by DHTs. In: Proceedings of the SIGCOMM, Pisa, Italy, 11-15 September, ACM, pp 351–362
10. Al-Mamou A, Labiod H (2007) ScatterPastry: An overlay routing using a DHT over wireless sensor networks. In: Proceedings of the international conference on intelligent pervasive computing, Jeju City, 11-13 October, IEEE, USA, pp 274–279
11. Al-mamou A, Schiller J, Labiod H, Mesut G (2007) A case for an overlay routing on top of MAC Layer for WSN. In: Proceedings of the second international conference on sensor technologies and applications (sensorcomm'2008), Cap Esterel, 25-31 August, IEEE, USA, pp 87-92
12. Fuhrmann T, Di P, Kutzner K, Cramer C (2006) Pushing chord into the underlay: Scalable routing for hybrid manets. Tech. Rep.
13. Muneeb A, Koen L (2007) A case for peer-to-peer network overlays in sensor networks. In: Proceedings of the international Workshop on Wireless Sensor Network Architecture (WWSNA) with 6th IPSN, Cambridge, Massachusetts, USA, pp 56–61
14. Sioutas S, Oikonomou K, Papaloukopoulos G (2009) Building an efficient P2P overlay for energy-level queries in sensor networks. In: Proceedings of the international conference on management of emergent digital EcoSystems (MEDES'09), Lyon, France, 27-30 October, ACM, New York, NY, USA
15. Fersi G, Louati W, Ben Jemaa M (2010) Energy-aware virtual ring routing in wireless sensor networks. J Netw Protocols Algorithms 2:16–29
16. Longxiang G, Ming L (2009) Multi-level virtual ring: A foundation network architecture to support peer-to-peer application in wireless sensor network. In: Proceedings of Australasian telecommunication networks and applications conference, Canberra, Australia, 10-12, November, IEEE, USA
17. NS2 website Available at http://www.isi.edu/nsnam/ns/
18. Christin D, Reinhardt A, Mogre P, Steinmetz R (2009) Wireless Sensor Networks and the Internet of Things: Selected Challenges. In: Proceedings of the 8th GI/ITG KuVS Fachgesprach, “Drahtlose Sensornetze”
19. Fersi G, Louati W, Ben Jemaa M (2013) The optimal transmitting power in randomly deployed heterogeneous Wireless Sensor Networks for predetermined average node degree. In: Proceedings of the 9th International Wireless Communications & Mobile Computing Conference (IWCMC 2013). Cagliari, Sardinia, Italy
20. Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H (2001) Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the ACM conference of the Special Interest Group on Data Communication, San Diego, CA
21. Rowstron A, Druschel P (2001) Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: Proceedings of IFIP/ACM international conference in distributed systems platforms, Heidelberg, Germany
22. Kleinrock L, Silvester JA (1978) Optimum transmission radii for packet radio networks or why six is a magic number. In: Proceedings of the IEEE national telecommunications conference, Birmingham, Alabama, Dec. 1978, pp 431–435
23. Takagi H, Kleinrock L (1984) Optimal transmission ranges for randomly distributed packet radio terminals. IEEE Trans Commun 32(3):246–257
24. Avidor D, Mukherjee S, Atay F (2007) Transmit power distribution of wireless ad hoc networks with topology control. In: Proceedings of the IEEE International Conference on Computer

Communications (INFOCOM'07), Anchorage, Alaska, USA, pp 46–52

25. Wang Y, Wang X, Agrawal DP, Minai A (2006) Impact of heterogeneity on coverage and broadcast reachability in wireless sensor networks. In: Proceedings of the international conference on computer communications and networks (ICCCN'06), Arlington, Virginia, USA, pp 63–67
26. Bettstetter C (2002) On the minimum node degree and connectivity of a wireless Multihop network. In: Proceedings of the ACM international symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'02), EPF Lausanne, Switzerland, June 2002, pp 80–91

Ghofrane Fersi graduated in Computer Science engineering from the National School of Engineers of Sfax (ENIS), Tunisia, in 2008. She received her M.S. degree (DEA) in Systems of Information and dedicated New Technologies (NTSID) in 2009, from ENIS. Since December 2010, she has joined the Research Unit of Development and Control of Distributed Applications (ReDCAD) at the National School of Engineering of Sfax (ENIS) as a Ph.D. candidate.

She received the PhD. degree in computer science from ENIS, University of Sfax in 2014.

Since 2013, she has been an Assistant Professor in the Department of computer science in the university of Sousse, Tunisia.

Her current research focuses on Distributed Hash Table-based protocols in Wireless Sensor Networks, Internet of Things protocols and architectures.

Wassef Louati graduated in Computer Science engineering from the National School of Engineers of Sfax,Tunisia, in 2003. In 2004, he joined the SAMOVAR (Distributed Services, Architectures, Modelling, Validation and Network Administration) research unit at the National Institute of Telecommunications (GET-INT), France, as a Ph.D. candidate. He received the Ph.D. degree in Computer Science from the Pierre et Marie Curie University (Paris 6) and GET-INT, in 2007. Since 2008, he has been an Associate Professor in the Department of Computer Science at the University of Monastir, Tunisia. His research activities are focused on Distributed Hash Table systems and applications in Wireless Sensor Networks.

Maher Ben Jemaa obtained his diploma of Engineer in Computer Science from the National School of Computer Sciences ENSI (Tunisia) in 1989 and his Ph.D. from the Department of Electrical Engineering at the National Institute of Applied Sciences (INSA) Rennes (France) in 1993. He joined the National School of Engineers of Sfax (ENIS) as Assistant Professor of Computer Science in the Department of Computer Science and Applied Mathematics in 1995. He became an Associate Professor in 1997 and he is a keynote professor since March 2011. He is actually the head of the Computer Science and Applied Mathematics in the National School of Engineers of Sfax (ENIS). His current research areas include Fault Tolerance of distributed systems, Quality of Service in Ad hoc Networks and routing issue in Wireless Sensor Networks.