

An Energy-Aware Clustering Scheme for Mobile Applications

Carmela Comito
DEIS, University of Calabria
Rende (CS), Italy
Email: ccomito@deis.unical.it

Domenico Talia
ICAR-CNR
DEIS, University of Calabria
Rende (CS), Italy
Email: talia@deis.unical.it

Paolo Trunfio
DEIS, University of Calabria
Rende (CS), Italy
Email: trunfio@deis.unical.it

Abstract—In a mobile ad hoc network, efficient routing, resource allocation, and energy management can be achieved through *clustering* of mobile nodes into local groups. In this paper we propose a clustering scheme that allows self-configuration and adaptation of the network, prolonging its lifetime by distributing energy consumption among clusters. A combined weighted metric approach is used to select cluster-head nodes, taking into account energy, mobility and location of nodes. We have evaluated the clustering scheme using a prototype of the system which includes smart phones and Android emulators. The experimental results show that our clustering approach results in high average nodes' residual life, proving its effectiveness in prolonging network lifetime.

Index Terms—Mobile computing; Energy efficiency; Clustering.

I. INTRODUCTION

A mobile ad hoc network (MANET) is a self-configuring network of mobile devices connected by ad hoc wireless links and equipped with networking capabilities. Recent developments in the technologies of laptops and PDAs together with the reduction of their costs have incredibly raised the interest in MANETs.

A key aspect to be addressed to enable effective and reliable computing over mobile devices is ensuring energy efficiency, as mobile devices are battery-power operated and lack a constant source of power. Most commercially available mobile computing devices like PDAs and mobile phones have battery power which would last for only a few hours. Therefore, the next generation of mobile applications for such mobile devices should be designed to minimize the energy consumption.

In a mobile ad hoc network, efficient routing, resource allocation, and energy management can be achieved through *clustering* of mobile nodes into local groups. Clustering the network promotes and eases collaborations among mobile users. Examples of *mobile-to-mobile* collaborations include different contexts such as disaster relief, healthcare, construction management where co-workers can collaboratively execute (in a peer-to-peer style) computational tasks. Whenever a resource limited device (client) in such a cooperative environment has a set of tasks to be executed it can use all available resources in nearby computing devices (servers).

In this paper we propose a clustering scheme where mobile devices are organized into local groups (clusters or mobile

groups). Each cluster has a node referred to as cluster-head that acts as the local coordinator of the cluster and is in charge of cluster formation and maintenance. The main design principles of our clustering scheme is to allow self-configuration and adaptation of the network, prolonging its lifetime by distributing energy consumption among clusters. To this aim, we designed and implemented a bottom-up cluster formation algorithm that finds successive clusters using previously established ones by maximizing the network residual life. We characterize the energy consumption of mobile devices by introducing a new energy efficiency model. A weighted metric that combines the effect of energy, mobility and location of nodes is introduced to select suitable cluster-head nodes.

We have evaluated the clustering scheme using a prototype of the system which includes smart phones and Android emulators. The experimental results show that our clustering approach introduces only a small overhead to the energy consumption and prove the benefits of clustering the network as it is effective in prolonging network lifetime.

The remainder of the paper is organized as follows. Section II discusses related work. Section III presents the energy model. Section IV describes the basics of our clustering scheme. The energy-aware clustering protocol and related algorithm are presented in Section V. Section VI presents the experimental results. Finally, Section VII concludes the paper.

II. RELATED WORK

In literature there are several approaches to cluster MANET networks. The clustering schemes can be classified according to different design principles criteria. Low-maintenance clustering schemes [12], [3] aim at providing stable cluster architectures with little cluster maintenance cost by limiting re-clustering situations or minimizing explicit control messages for clustering. Mobility-aware clustering [11], [4], [5] takes the mobility behavior of mobile nodes into consideration. By grouping mobile nodes with similar speed into the same cluster, the cluster structure can be correspondingly stabilized. This approach minimizes the number of cluster-heads but due to change in the network topology it may produce high cluster-head re-elections. Energy-efficient clustering [6], [7], [8] aims to use the battery energy of mobile nodes more efficiently by balancing energy consumption among different mobile nodes.

In this way the network lifetime can be prolonged. Load-balancing clustering schemes [6], [7], [8], [9] attempt to limit the number of mobile nodes in each cluster to a specified range so that clusters are of similar size. Thus, the network loads can be more evenly distributed in each cluster. Combined metrics based clustering (WCA) [10] considers multiple metrics, such as node degree, cluster size, mobility speed, and battery energy, in cluster configuration, especially in cluster-head selections. Taking into account more parameters, cluster-heads can be more properly chosen. Moreover, the weighting factor for each parameter can be adaptively tuned according to different application scenarios.

The proposed clustering scheme falls under the combined metrics based approach. From the above discussion, we can see that most clustering algorithms, except the WCA, use only one metric. This is limiting because only by considering the combined effect of key parameters it is possible to catch the global behavior of a node. The WCA approach [10] is the mostly related to us. However, we use different parameters. Instead of the battery power we use the residual life because it considers not merely the remaining energy of the device but also the energy load of the node. We consider also the number of neighbors in addition to the degree difference. In contrast, WCA chooses as candidate cluster-heads only those nodes whose number of neighbors is under a given threshold, thus precluding to nodes with good energy level, coverage range and mobility characteristics the possibility of becoming cluster-heads. Finally, we also use as metric the transmission range because a candidate cluster-head has to interact with other clusters and hence must ensure a good coverage area. Differently to most of the related approaches where the clustering protocol is invoked periodically introducing high overhead, our clustering scheme is invoked only when needed: (i) a new node joins the network or (ii) addressing emergencies like energy drains or node unreachability due to its mobility. Therefore, our clustering scheme produces a cluster infrastructure with minimized clustering-related maintenance cost.

III. ENERGY MODEL

Mobile nodes are battery powered [1], which makes energy a critical concern. Thus, the main aim in MANET networks is to conservatively consume the energy in order to increase the lifetime of the network.

Energy consumption of mobile devices depends on the computation and the communication loads. We define E_i as the rate of energy consumption of node i in a time interval δt , which is the sum of all energy consumption for communication, ET_i , and computation, EC_i , of all the tasks assigned to node i within the time interval δt :

$$E_i = EC_i + ET_i \quad (1)$$

Our approach is to estimate the energy consumption for computation and to analytically evaluate the energy consumed for communication. This last issue is the main aim of the section.

In the following some definitions are introduced to support the proposed energy model.

Definition 1: Let $RE_i(t)$ be the residual energy available at node i at time t , and $P_i(t)$ the instantaneous power; the residual life of node i at time t , $RL_i(t)$, is defined as follows:

$$RL_i(t) = RE_i(t)/P_i(t). \quad (2)$$

According to [2], we assume a commonly used wireless propagation model where the received signal power attenuates proportionally to $TR^{-\alpha}$, where TR is the transmission range and α is the loss constant, typically between 2 and 4 depending on the wireless medium. Based on this model, we can introduce the definition of a wireless link. A wireless link exists between two mobile devices if the transmitting node transmits with sufficiently high power such that the signal-to-interference-plus-noise-ratio (SINR) at the receiving node is greater than a given threshold value δ . The threshold value δ is chosen to achieve a desired bit-error-rate for the given modulation scheme and data rate. In other words, a wireless link can be established between two nodes i and j if the distance between them, denoted as r_{ij} is within the transmission range of the sender node, that is $r_{ij} \leq TR_i$.

Definition 2: The neighbors of a node i are the nodes falling in its transmission range TR_i . As such, a wireless link can be established between node i and each of its neighbors. The number of active wireless links of a node i is denoted as the *degree* of node i .

To determine the energy consumed for communication by a node is necessary to distinguish the node state. The network interface of a mobile device can be in four states: (i) transmit mode; (ii) receive mode; (iii) idle mode (this is the default mode for ad-hoc network and in this state a node can transmit or receive); (iv) sleep mode (characterized by really low power consumption). In this state the interface can neither transmit nor receive until it is woken up and changes state. In MANET networks, nodes must always be ready to receive traffic from neighbors due to the absence of base station nodes. Thus, a network interface operating in ad-hoc mode can not be in a sleep mode but it has to continuously listen to the wireless channel consuming this way a constant idle energy power. Therefore, every node *overhears* every packet transmission occurring in its transmission range consuming this way energy uselessly. Thus, the idle energy consumption is referred to as *overhearing*. Due to the overhearing, a new cost in the computation of per-packet energy consumption is introduced and it is the cost for discarding overheard packets. Therefore, to model the energy consumed for communication, the costs to send, receive and discard a packet must be included. Consequently, the energy consumed by a node i for communication can be defined by the following equation:

$$ET_i = E_{\text{send}_i} + E_{\text{receive}_i} + E_{\text{discard}_i} \quad (3)$$

A packet may be sent through a broadcast or a point-to-point channel. With the former mode the packet is received by all hosts within the sender's transmission range; whereas with the latter mode it is discarded by non-destination hosts.

We refer to a simple radio model where the transmitter dissipates energy (i) to turn the radio electronics, E_{elec} , for digital coding, modulation, filtering, spreading of the signal and (ii) for the power amplifier, E_{amp} . On the contrary, the receiver dissipates energy only to run the radio electronics. Based on this model, the cost, $E_{send_{ij}}$, for a node i to send a point-to-point packet to a node j is described by the following equation:

$$E_{send_{ij}} = |MSG|(E_{elec} + r_{ij}^\alpha * E_{amp}) \quad (4)$$

where $|MSG|$ is the size (number of bits) of the message exchanged among nodes i and j . In the case broadcast transmission is used, to send a packet it is required a power level necessary to reach the most faraway node (denoted as j_{max}) among the ones within the sender's transmission range and Equation 4 changes as follows:

$$E_{broad_i} = |MSG|(E_{elec} + r_{ij_{max}}^\alpha * E_{amp}) \quad (5)$$

Thus, the aggregate energy cost for sending packets (communicating either data or just synchronization messages) depends on the transmission mode used. If point-to-point transmission is used, the energy consumption is the sum of the costs on all communication links having i as transmitting node, otherwise the energy consumption is due to the broadcast cost. If s_{pp} are the active point-to-point transmitting links of node i , the overall sending cost of node i is given by the following equation:

$$E_{send_i} = \begin{cases} \sum_{j=1}^{s_{pp}} E_{send_{ij}} & \text{if point-to-point} \\ E_{broad_i} & \text{otherwise} \end{cases} \quad (6)$$

The aggregate energy cost of a node i for receiving, $E_{receive_i}$, is the sum of all the costs on all communication links having i as receiving node. If there are k communication links, the overall cost for receiving is:

$$E_{receive_i} = k * |MSG| * E_{elec} \quad (7)$$

As said before, a network interface overhears all traffic sent and received by nearby nodes. Thus, it is important to consider the energy consumed during the processing of point-to-point traffic due to packet discarding by the non recipients of those packets. Non destination nodes within the transmission range of either the transmitting or receiving nodes overhear the traffic. The cost of discarding is comparable to the cost of receiving. Therefore, if the number of contemporary discardings is n_{disc} , the overall cost for discarding of a node i is given by:

$$E_{discard_i} = n_{disc} * |MSG| * E_{elec} \quad (8)$$

IV. THE CLUSTERING SCHEME

In a wireless mobile ad hoc network, which changes its topology dynamically, efficient resource allocation, energy management and routing can be achieved through adaptive *clustering* of the mobile nodes.

In a clustering scheme the mobile nodes are divided into virtual groups. Generally, geographically adjacent devices are assigned to the same cluster. Under a cluster structure, mobile

nodes may be assigned a different function, such as *cluster-head* or *cluster member*. A cluster-head normally serves as the local coordinator for its cluster, performing intra-cluster transmission arrangement, data forwarding, and so on. A cluster member is usually called an ordinary node, which is a non cluster-head node without any inter-cluster links.

We refer to the system architecture depicted in Figure 1, designed to allow on-demand collaborations among mobile nodes. Examples of mobile-to-mobile collaborations occur in several domains such as disaster relief, construction management and healthcare. In order to promote and ease collaborations when two or more mobile users, who are members of the same organization or simply collaborate, meet each other, we let them grouping into clusters referred to as *mobile groups*. Consequently, the proposed architecture includes some stationary nodes and a number of mobile groups or clusters. Figure 1 shows the interactions among

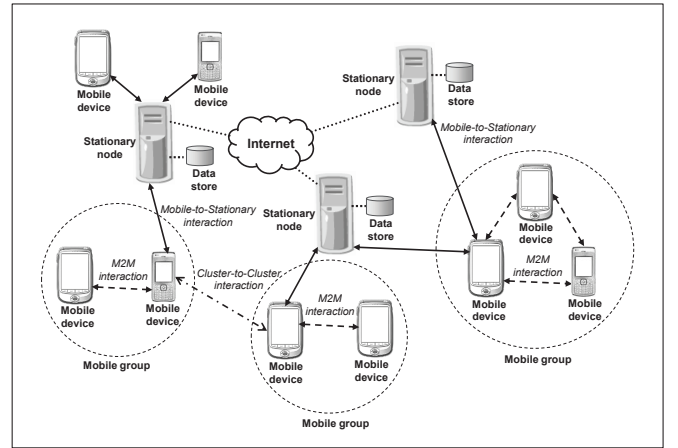


Fig. 1. The system architecture. The arrows denote remote service calls.

the different components of the architecture. Stationary nodes are connected through the Internet and can interact with the other nodes (including the mobile ones) in order to execute a computational task. Mobile nodes within a group interact through ad-hoc connections (e.g., wi-fi, bluetooth) that we refer to as *M2M connections*, represented as dotted arrows in Figure 1. Interactions among mobile groups (*cluster-to-cluster connections*) take place through ad-hoc connections among the cluster-heads of the groups and are represented as dot-dash arrows. Mobile groups are connected to stationary nodes through their cluster heads (*mobile-to-stationary connections*) by exploiting an Internet connection (e.g, wi-fi, wi-max). All types of interactions take place either to ask for a computation request or to cooperate in order to collaboratively execute a computational task.

The proposed clustering scheme is based on a fully distributed cluster formation algorithm in which nodes take autonomous decisions; no global communication is needed to setup the clusters but only local decisions are taken autonomously by each node. This means that the proposed architecture is self-organized into mobile clusters: when mobile devices meet each other, i.e., they are within the same

transmission range, they can form a mobile group. The self-organization nature of the clustering scheme distributes the responsibility among the different mobile nodes. In such a way, there is no node in charge of the overall organization; each individual node interacts directly with the other ones in its transmission range in a peer-to-peer fashion.

In the design of the clustering scheme we made the following assumptions. (1) The number of mobile nodes in the network changes as nodes dynamically change their position. (2) A given node can belong only to one cluster at a given time. (3) There are no fixed cluster-head nodes in the cluster formation process.

Choosing cluster-heads is the key operation in the clustering scheme. The number of cluster-heads depends on many factors like the number of nodes in network, their physical location, the transmission range and the energy level. To select the cluster-head nodes we introduce a combined weighted metric that takes into account the following node parameters:

- *Residual life (RL)*. According to the energy model introduced above, a node with higher residual life corresponds to a less loaded node and thus to a more powerful node that is a good candidate to be elected as a cluster-head.
- *Transmission range (TR)*. A node with higher transmission range means a larger covered area. A large coverage area is an important requirement that a candidate cluster-head node should meet since it acts also as a gateway towards the other clusters in the network.
- *Neighbors (NB)*. The size of each local group also depends on the number of nodes within the cluster-head range. The larger the number of neighbors of a node, the greater the likelihood of being elected cluster-head.
- *Mobility (M)*. A node with lower mobility has a higher chance of being a cluster-head. According to [10], we estimate node mobility by taking the average of the distances covered by it in the last t periods of times. This gives a prediction of node mobility characteristics. Thus, we introduce a mobility parameter M defined by the following equation:

$$M_i(t) = 1/t \sum_{j=1}^t \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2} \quad (9)$$

where (x_j, y_j) and (x_{j-1}, y_{j-1}) are the coordinates of node i at time j and $j - 1$.

Depending on the specific application, different subsets of the parameters above can be used in the metric to elect the cluster-heads. Thus, to establish whether a node i can be elected as a cluster-head, a *Cluster-Head Selection* function, $CHS_i(t)$, is defined. Such a function is described by the following equation:

$$CHS_i(t) = \alpha RL_i(t) + \beta/M_i(t) + \gamma TR_i + \delta NB_i(t) \quad (10)$$

where α , β , γ and δ are the weights corresponding to the above cited performance parameters.

The node within a cluster having the maximum value of the CHS function will be selected as the cluster-head. The weighting factors are chosen such that $\alpha + \beta + \gamma + \delta = 1$.

Therefore, the contribution of each parameter in the CHS function can be tuned by selecting suitable combinations of the weighting factors. For example, in a particular system configuration where the residual life of the device is more important, the weight associated to this parameter can be set to a larger value. On the other hand, the mobility factor and the battery power would be the same for all the nodes when the system is initialized. Thus, the mobility parameter and the residual life will not be taken into consideration in the CHS function when the system is initially set up. Moreover, in the case of a system with uniform mobile devices the transmission range will be the same for all the devices, thus this parameter will have a weighting factor $\gamma = 0$ and will not appear in the CHS function.

V. AN ENERGY-AWARE CLUSTERING PROTOCOL

Using the proposed weighted approach, in this section we describe an energy-aware clustering protocol that consists of two main phases: (i) *cluster formation* and (ii) *network maintenance*.

A. Cluster formation

We propose a simple clustering formation algorithm that configures clusters to extend the network lifetime. The algorithm begins with each device as a separate cluster if no already established clusters are found in its transmission range. Otherwise, it allows the incoming node to join a cluster on the basis of energy constraints. In particular, the algorithm executed by a node to join the network is described as follows.

A mobile node i that wants to join the network has to establish whether it can join an already existing cluster or it has to create a new one. To this aim, node i sends a join request message to advertise its presence and to check the presence of other nodes within its transmission range. One of the two events may occur:

(i) Node i does not receive any reply. In this case it will form a new group electing itself as the cluster-head of the group.

(ii) Node i receives one or more replies. If node i has more than a group within its transmission range it will receive more responses. In this case, to balance the energy load over all the nodes in the network, node i will choose the cluster that will allow to extend the network lifetime. In particular, node i will join the local group that maximizes the life of the whole network as expressed by the following equation:

$$\text{Max} \sum_{j=1}^N \alpha_j RL_{LG_j}(t) \quad (11)$$

where RL_{LG_j} denotes the residual life of local group LG_j and N is the number of groups in the network. The residual life of each local group is described by the following equation:

$$RL_{LG_i} = \sum_{j=1}^{N_{LG_i}} \alpha_j RL_j(t) \quad (12)$$

where N_{LG_i} is the number of nodes within local group LG_i , RL_j is the residual life of node j in the group, and parameter

α_j takes into account the importance of node j in the local group.

The algorithm followed by a node to join the network and, thus, to select its cluster-head is described in Figure 2. We use a state diagram formalism to model the behavior of a generic node (mobile device), which includes the behavior of nodes performing join requests (NOT_CLUSTER_MEMBER nodes), as well as the behavior of nodes responding to join requests (CLUSTER_HEAD nodes). The state diagram is represented using a pseudo-code that describes states, macro-states (containers for other states), and events that produce transitions among states.

As shown in Figure 2, each node includes three fields: the identifier (MY_ID), the residual life (MY_RL), and the identifier of its cluster head (CH_ID). Note that CH_ID may be *null* (cluster head not yet chosen) or equal to MY_ID (the node itself is a cluster head). As soon as a node powers on, it transits to the NOT_CLUSTER_MEMBER macro-state, as specified by the *entry* section of the NODE state diagram.

A node in the NOT_CLUSTER_MEMBER macro-state possesses a list JRS that will be used to contain the responses to its join requests. The *entry* section of this macro-state initializes CH_ID and JRS, broadcasts a *JoinRequest* message containing its identifier and residual life, and transits to the internal state RCV_JOIN_RESPONSE.

When the node is in the RCV_JOIN_RESPONSE state, two events may happen: *i*) a *JoinResponse* message is received from a cluster-head, in which case the message received is added to the JRS list; *ii*) an internal timeout expires, which causes a transition to the CH_SELECTION state.

As soon as the CH_SELECTION state is reached, a *selectCH* operation is invoked to obtain the identifier of the best cluster-head (CH_ID) based on the join responses collected into the JRS list. According to Equation 11, the best cluster-head to join is the one allowing to maximize the network lifetime. If CH_ID is equal to MY_ID, then the node will be the cluster-head of a cluster including only the node itself, causing a transition to the CLUSTER_HEAD state. Otherwise, the node sends a *JoinAck* message to the cluster-head, and transits to a CLUSTER_MEMBER state (not shown in Figure 2).

A node in the CLUSTER_HEAD macro-state possesses a list of cluster members (CMS), and the residual life of the headed cluster (CL_RL). The *entry* section of this macro-state causes a concurrent transition to the JOIN_MGMT macro-state, where join requests from new nodes are managed, and to other macro-states (not shown here) where other management tasks are performed, like cluster aggregations, cluster-head rotations, and so on.

Nodes in the JOIN_MGMT macro-state possess a field RQ_ID used to store the identifier of the node whose join request is currently being managed. The *entry* section initializes to *null* the RQ_ID field (i.e., no request is being managed) and produces a transitions to the RCV_JOIN_REQUEST state.

When the cluster head is in the RCV_JOIN_REQUEST state, it can receive a *JoinRequest* message. In this case the following operations are performed: RQ_ID is set to the

identifier of the requester; an *estimateRL* operation is invoked to estimate the residual life of the cluster (CL_ERL) on the basis of the current value of CL_RL and the residual life of the requester; sends a *JoinResponse* message that includes the identifier of the cluster-head, the current value of CL_RL, and CL_ERL; transits to the RCV_JOIN_ACK state.

Two events can happen in the RCV_JOIN_ACK state: *i*) a *JoinAck* is received by RQ_ID, in which case that node is added to the CMS; *ii*) an internal timeout expires, which causes a transition to the RCV_JOIN_REQUEST state. Note that each cluster-head processes only one *JoinRequest* at time; the timeout in the RCV_JOIN_ACK determines the amount of time the cluster-head will wait for a possible *JoinAck* before returning to the RCV_JOIN_REQUEST state.

The node join algorithm is also executed to let nodes (re-)affiliate to clusters. The nodes in the network forming a single-member cluster periodically invoke the clustering formation algorithm to assess whether they can join another cluster. Moreover, sometimes a node has to be re-affiliated because the signal strength received from the cluster-head decreases under a given threshold, and the node cannot stay connected to that cluster-head. This means that the mobile device is moving out of the range of the group it belongs to, and so it becomes unreachable by the other nodes in the group. In such a case, a re-affiliation is needed: the node can join another cluster following the cluster formation algorithm described above.

B. Cluster maintenance: Cluster-head re-election

Cluster-head re-elections may occur due to different reasons. It is often the case that the cluster-head re-election affects only nodes within a cluster without causing any re-affiliations but only changing the cluster-head role from a node to another one of the same cluster. In this case the re-election consists of a *cluster-head rotation* within the group whereas in some other cases node *re-affiliations* are necessary during the cluster-head re-election process. More precisely, the cluster-head re-election process takes place whether one of the following events occurs.

To expand the life of a group. Periodically it could be necessary to perform a rotation of the cluster-head role within the group to expand the group lifetime. This reduces the number of re-affiliations lowering the cluster maintenance cost. In particular, a periodical check over the current cluster-head could be performed (i) to evenly distribute the load among the nodes within a cluster; (ii) to avoid that the cluster-head lifetime decreases under a given threshold; (iii) because at a certain point another node in the cluster has a value of CHS function better than the one of the cluster-head. Thus, the cluster-head re-election process is activated either periodically or when the current cluster-head residual life is under a given threshold value. To this aim the cluster-head maintains a *balancing* threshold and when its residual life reaches this value the cluster-head re-election algorithm is executed. The algorithm selects the node within the group, among the ones reaching all the others cluster members, with higher CHS

```

1 StateDiagram NODE {
2   NodeID MY_ID;
3   Float MY_RL;
4   NodeID CH_ID;
5   entry() {
6     transition to NOT_CLUSTER_MEMBER;
7   }
8   Macro-State NOT_CLUSTER_MEMBER {
9     List JRS;
10    entry() {
11      CH_ID = null;
12      JRS = empty;
13      broadcast JoinRequest(MY_ID,MY_RL);
14      transition to RCV_JOIN_RESPONSE;
15    }
16    State RCV_JOIN_RESPONSE {
17      on event JoinResponseReceived(JoinResponse
18        res) {
19        JRS.add(res);
20      }
21      on event TimeoutExpired() {
22        transition to CH_SELECTION;
23      }
24    }
25    State CH_SELECTION {
26      entry() {
27        CH_ID = selectCH(JRS);
28        if (CH_ID == MY_ID) {
29          transition to CLUSTER_HEAD;
30        }
31        else {
32          send JoinAck(MY_ID) to CH_ID;
33          transition to CLUSTER_MEMBER;
34        }
35      }
36    }
37  }
38  Macro-State CLUSTER_HEAD {
39    List CMS;
40    Float CL_RL;
41    entry() {
42      concurrent transition to JOIN_MGMT and <other
43        management macro-states>
44    }
45    Macro-State JOIN_MGMT {
46      NodeId RQ_ID;
47      entry() {
48        RQ_ID = null;
49        transition to RCV_JOIN_REQUEST;
50      }
51      State RCV_JOIN_REQUEST {
52        on event JoinRequestReceived(JoinRequest
53          req) {
54          RQ_ID = req.ID;
55          Float CL_ERL = estimateRL(CL_RL,req.RL);
56          send JoinResponse(MY_ID,CL_RL,CL_ERL) to
57            RQ_ID;
58          transition to RCV_JOIN_ACK;
59        }
60      }
61      State RCV_JOIN_ACK {
62        on event JoinAckReceived(JoinAck ack){
63          if(ack.ID == RQ_ID) {
64            CMS.add(RQ_ID);
65          }
66        }
67      }
68      on event TimeoutExpired(){
69        transition to RCV_JOIN_REQUEST;
70      }
71    }
72  }

```

Fig. 2. Node join algorithm for cluster formation.

value. However, if none of the cluster members satisfies the reachability condition, the cluster-head re-election algorithm fails and the cluster-head keeps its role.

Cluster-head drains out of battery power. The *balancing* threshold is not an emergency threshold, thus nothing will happen if the rotation is not feasible. On the contrary, when the cluster-head is draining out of energy it can no longer act as cluster-head. Therefore, in this case the cluster-head tries to activate the re-election process. However, due to its limited residual energy, the process can fail. The cluster-head disappears and its members have to be re-affiliated by executing the clustering formation algorithm.

VI. PERFORMANCE EVALUATION

In this section we present a simulation study of our clustering scheme. We performed the experiments over a network composed by a set of smart phones and Android emulators. Unless otherwise specified, the parameters used in the simulation are as follows. For the CHS function, we used the following weights: $\alpha = 0.4$, $\beta = 0.4$, $\gamma = 0.0$ and $\delta = 0.2$. The high values for α and β take into account the importance of residual life and mobility in cluster-head selection; the zero value for γ has been set because all the devices used in our simulation are homogeneous, therefore their transmission range are identical; finally, we assigned the residual weight to δ to take also into account the number of neighbors of each node. The simulation area was set to 250,000 m². The initial value of the transmission range was set to 100 meters and the initial energy level on each device is 24300 J. Each device was equipped with a network interface 802.11 b/g, with a bandwidth of 11 Mbps. In all the experiments, we refer to the computation load as the energy consumption, a priori estimated, to execute a set of sample tasks.

In the simulation environment the nodes could move in all possible directions with displacement varying uniformly between 0 to a maximum value per time unit. The mobility model considered for the simulation is *random way point* [13]. According to this model, each node begins by pausing in one location for about 90 seconds and then chooses a uniformly distributed random speed, ranging from 5 to 35 m/s, and destination and moves to that destination in a straight line with that selected speed.

As the aim of our clustering scheme is optimizing the energy consumption in the network, the simulation aims to study the behavior of the framework with respect to the energy perspective and thus lifetime of the whole network. We also evaluated the topological characteristics of the network.

A. Energy efficiency

By means of this set of experiments we want to evaluate the energy consumption behavior of our clustering scheme. Since our clustering scheme tries to maximize the energy levels of the nodes and thus maximize the network lifetime, our metrics measure the residual life of the nodes and of the whole network. As our clustering scheme attempts to balance the load over different cluster-heads, we expect that the load balancing prolongs the lifetime of the network by enhancing the average energy levels.

In a first experiment we measured the average residual life of cluster-heads with respect to the simulation time. We considered homogeneous cluster-heads with an initial energy value equals to 22900 J, degree 4, computational load of 1 J/s, packet size of 512 Bytes, and traffic bound to a packet every 20 seconds. Figure 3 shows that cluster-head nodes have high average residual life. In particular, up to 200 minutes of simulation the average residual life does decrease very slowly

with time. Then, the residual life decreases faster but remains linear. After 350 minutes of simulation on average the cluster-head residual life reaches the 20% of its initial value and if the cluster-head re-election algorithm is not invoked the network will expire after 380 minutes of simulation. Thus, another key aspect of our clustering scheme is that cluster-heads change their role based on their energy value, and so on average the residual life of the cluster-heads does not reach zero within the period simulated.

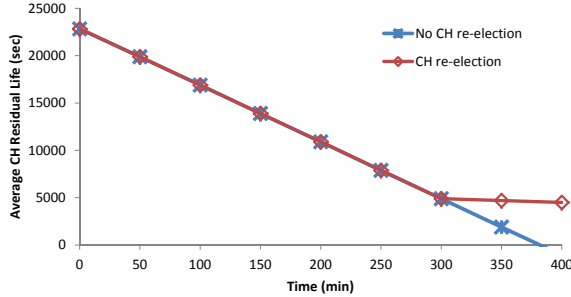


Fig. 3. Average CH Residual Life w.r.t. simulation time.

Dually, the average energy dissipated by the cluster-heads in the system does increase linearly with the simulation time as shown in Figure 4. As for the residual energy, if the cluster-head re-election algorithm is not activated the node will consume 100% of its initial energy.

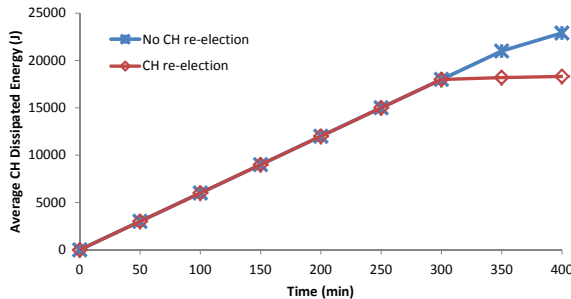


Fig. 4. Average CH Dissipated Energy w.r.t. simulation time.

With another set of experiments we want to study how the residual life of the nodes is influenced by the energy consumption. As the energy cost is determined by the communication and computation loads, we want to assess how these two cost items contribute to the residual life of the nodes. To this aim, we first evaluated how the residual life changes with the computation load. We consider homogeneous cluster-head nodes with remaining energy equal to 22113 J, degree 10, traffic of a packet every 20 seconds, packet size of 512 Bytes. The experimental evaluation confirms our theoretical expectation as we can see in Figure 5 where the two histograms are almost coincident. The residual life does decrease with the computation load. The trend of the graph is logarithmic, hence proving the high influence of the computation load.

We also evaluated the impact of the computation load on the network lifetime. As we can see from Figure 6, with

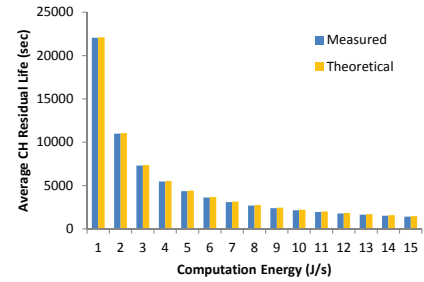


Fig. 5. Average CH Residual Life w.r.t. computation load.

the increasing of the computation load the network lifetime becomes shorter. With a computation load of 30 J/m the network lifetime is 800 minutes, whereas when the computation load is 120 J/m the network lifetime decreases by 50%. Thus, only for long simulation time and only when the computation energy is high, the network life expires. It is worth noting that the network lifetime does decrease very slowly with the simulation time particularly when the energy load is not so high. Nevertheless, our clustering scheme allows to maintain a very high network lifetime.

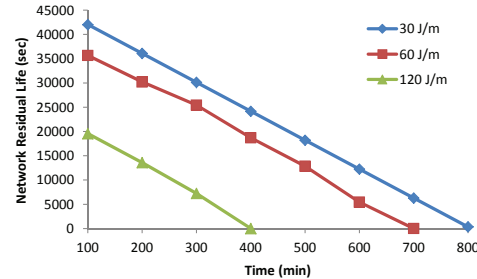


Fig. 6. Network Residual Life w.r.t. simulation time for different computation loads.

As expected, the evaluations show that the computation load impacts on the network lifetime. Conversely, the communication load marginally affects the life of the network as confirmed by the following experiment. We evaluated the impact of the packet traffic on the life of the network where 60% of nodes sends packets to their cluster-head with a rate that varies from a packet every 5 seconds to a packet every second. The simulation time is 100 minutes, the computation load is fixed to 0.5 J per second and the packet size is 128 kB. As expected, the increasing of the communication energy only slightly impacts on the network residual life as shown in Figure 7 where the network lifetime remains almost constant with the transmission rate.

To assess the overhead introduced by the clustering scheme, we measured the energy consumed for the clustering on the total energy dissipated in the system. As it is evident from Figure 8, the overhead introduced by the clustering scheme is really marginal and the energy consumed for clustering set up and management is negligible, representing only a very little fraction of the overall energy dissipated in the system.

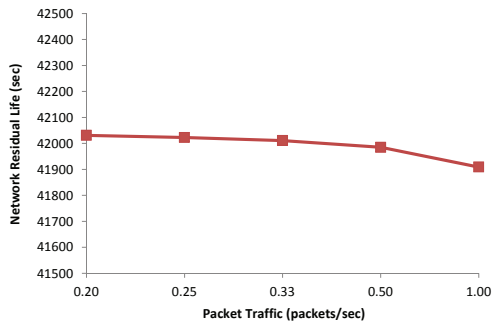


Fig. 7. Network Residual Life w.r.t. transmission rate.

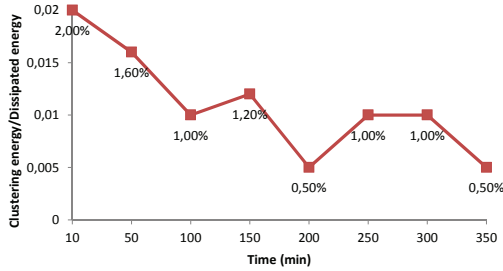


Fig. 8. Impact of the clustering energy on the overall dissipated energy.

B. Cluster characteristics

By this second set of experiments we analyzed the behavior of our clustering scheme in terms of the network topology characteristics such as number of clusters and number of single-node clusters.

Assuming a uniform distribution of nodes in the network, we evaluated how the number of clusters and thus the number of cluster-heads varies with the transmission range. Figure 9 shows that the percentage of cluster-head nodes decreases as the transmission range increases. In particular, the percentage is very high for small transmission ranges (60%) and it decreases rapidly with higher transmission ranges.

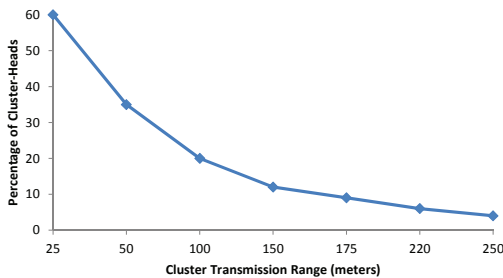


Fig. 9. Percentage of Cluster-Heads in the network.

Another important property of our clustering approach is the minimization of clusters composed of a single node through successive (re)-affiliations. Figure 10 shows that in our clustering scheme the percentage of clusters with more than one node is quite high: 70% for a transmission range of 50 meters, reaching 99% when the transmission range increases.

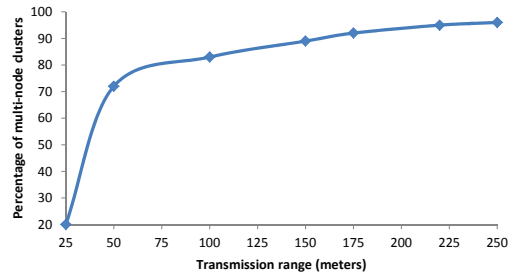


Fig. 10. Percentage of multi-node clusters in the network.

VII. CONCLUSION

In this paper, we have presented a clustering scheme for mobile computing focusing on energy efficiency. To conservatively consume energy and increase network lifetime we have introduced an energy-aware adaptive distributed clustering scheme based on a combined weighted metric. We have implemented a prototype of the system starting from the implementation of the software components devoted to cluster formation and energy measurements. An experimental evaluation has been performed to measure the network lifetime and the average dissipated energy by varying several parameters such as simulation time, transmission range, computation load, packet traffic. Results show that the proposed scheme produces high cluster-head residual life and thus a high network lifetime even after a long simulation time.

REFERENCES

- [1] L.D. Fife and L. Gruenwald, "Research Issues for Data Communication in Mobile Ad-Hoc Network Database Systems". *ACM SIGMOD RECORD*, 32(2):42–47, (2003).
- [2] T.S. Rappaport. "Wireless Communications: Principles and Practices". 2nd Edition, Prentice Hall, (2002).
- [3] J. Y. Yu and P. H. J. Chong. "3hBAC (3-hop between Adjacent Cluster-heads): a Novel Non-overlapping Clustering Algorithm for Mobile Ad Hoc Networks". *IEEE Pacrim03*, (1), pp. 318–21, (2003).
- [4] P. Basu, N. Khan, and T. D. C. Little. "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks". *IEEE ICDCSW'01*, pp. 413–18, (2001).
- [5] A. B. McDonald and T. F. Znati. "Design and Performance of a Distributed Dynamic Clustering Algorithm for Ad-Hoc Networks". *34th Annual Simulation Symp*, pp. 27–35, (2001).
- [6] A. D. Amis and R. Prakash. "Load-Balancing Clusters in Wireless Ad Hoc Networks". *3rd IEEE ASSET'00*, pp. 25–32, (2000).
- [7] J. Wu et al. "On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks". *J. Commun. and Networks*, 4(1):59–70, (2002).
- [8] J.-H. Ryu, S. Song, and D.-H. Cho. "New Clustering Schemes for Energy Conservation in Two-Tiered Mobile Ad-Hoc Networks". *IEEE ICC'01*, 3:862–66, (2001).
- [9] T. Ohta, S. Inoue, and Y. Kakuda. "An Adaptive Multihop Clustering Scheme for Highly Mobile Ad Hoc Networks". *Int. Symp. on Autonomous Decentralized Systems (ISADS'03)*, pp. 293–300, (2003).
- [10] M. Chatterjee, S. Das, and D. Turgut. "Wca: A weighted clustering algorithm for mobile ad hoc networks". *Cluster Computing Journal*, 5(2):193–204, (2002).
- [11] A. B. McDonald and T. F. Znati. "A Mobility-based Frame Work for Adaptive Clustering in Wireless Ad Hoc Networks". *IEEE JSAC*, 17:1466–87, (1999).
- [12] C. R. Lin and M. Gerla. "Adaptive Clustering for Mobile Wireless Networks". *IEEE JSAC*, 15:1265–75, (1997).
- [13] D. B. Johnson and D. A. Maltz. "Dynamic Source Routing in Ad-Hoc Wireless Networks". *Mobile Computing*, T. Imielinski and H. Korth, Eds, Kluwer, pp. 15381, 1996.