An Energy-efficient Clustering Solution for Wireless Sensor Networks

Dali Wei, Member, IEEE, Yichao Jin, Member, IEEE, Serdar Vural, Member, IEEE, Klaus Moessner, Member, IEEE, and Rahim Tafazolli, Member, IEEE

Abstract—Hot spots in a wireless sensor network emerge as locations under heavy traffic load. Nodes in such areas quickly deplete energy resources, leading to disruption in network services. This problem is common for data collection scenarios in which Cluster Heads (CH) have a heavy burden of gathering and relaying information. The relay load on CHs especially intensifies as the distance to the sink decreases. To balance the traffic load and the energy consumption in the network, the CH role should be rotated among all nodes and the cluster sizes should be carefully determined at different parts of the network.

This paper proposes a distributed clustering algorithm, Energy-efficient Clustering (EC), that determines suitable cluster sizes depending on the hop distance to the data sink, while achieving approximate equalization of node lifetimes and reduced energy consumption levels. We additionally propose a simple energy-efficient multihop data collection protocol to evaluate the effectiveness of EC and calculate the end-to-end energy consumption of this protocol; yet EC is suitable for any data collection protocol that focuses on energy conservation. Performance results demonstrate that EC extends network lifetime and achieves energy equalization more effectively than two well-known clustering algorithms, HEED and UCR.


I. INTRODUCTION

One of the key challenges of Wireless Sensor Networks (WSN) is the efficient use of limited energy resources in battery operated sensor nodes. Hierarchical clustering [1], [2], [3], [4], [5] has been shown to be a promising solution to conserve sensor energy levels [6], [7], besides being an effective solution to organizational tasks. With Cluster Heads (CH) that act as local controllers of network operations, a clustered WSN has an easily manageable structure.

A. Cluster Heads (CH)

The set of CHs in a WSN forms its backbone, providing a scalable solution to various networking tasks, such as data collection and habitat monitoring. At each cluster, a CH is responsible for various tasks, e.g. node association, authentication, and task assignment. The CH also maintains the cluster structure when node-centric events occur, such as hardware failures and sensor mobility. Support for traffic sharing, cluster membership, and inter-cluster connectivity are provided by collaborative discussions over the inter-CH links of the network backbone. Therefore, as a central control point of a cluster, a CH has considerably higher energy consumption compared to cluster members. This requires that the high load of CHs be distributed among all nodes.

B. Traffic hot-spots

Periodic reassignment of the CH role to different nodes helps prevent the problem of a single point of failure in the event of node energy depletion. However, traffic hot-spots [8], [9] in a WSN also pose error-prone situations. This is particularly important since clustered WSNs [10], [11], [12], [13] are mainly focused on data gathering applications (e.g. habitat monitoring and military surveillance), which involve periodic delivery of sensory data over multihop routes, creating highly congested areas, especially at locations close to a data sink (e.g. a control centre). Furthermore, there may also be other critically-located sensors not necessarily close to data sinks, which carry the burden of relaying large amounts of data traffic, especially when multiple high-rate routes pass through these sensors. Such nodes are usually frequently chosen to be data relays by routing algorithms and may serve a large portion of the network traffic, due to their convenient locations. Thus, avoiding the failure of such nodes caused by early energy depletion is critical to ensure a sufficiently long network lifetime.

C. Our Contribution

The hot-spot issue is particularly significant around sink nodes where large amounts of data are merged. In fact, as the hop distance to a sink decreases, the load on relay nodes quickly intensifies. Hence, there is an obvious relationship between the hop-distance to a data sink and the amount of data that has to be relayed. To obtain a well-balanced network load, this relation should be studied analytically. In doing so, the energy consumption of data communication and of control overhead caused by route discovery and any other procedures should be taken into account. In this paper, we argue that a node clustering solution can achieve this objective. We propose a scalable, distributed, and energy-aware clustering algorithm, Energy-efficient Clustering (EC). EC determines suitable cluster sizes considering their hop distances to the data sink. By tuning the probability that a node becomes a CH, EC effectively controls cluster sizes, which allows an approximately uniform use of the overall energy resources of a
WSN. In order to evaluate EC’s performance, we additionally propose an energy-efficient multihop WSN data collection protocol and calculate its energy consumption amounts. This protocol targets at low signalling overhead and an overall low level of energy consumption. Hence, EC can better conserve energy levels using the proposed protocol. However, EC is independent from the underlying data collection protocol and is adaptable to any data delivery protocol used for data collection to a sink node.

D. Paper outline

In the remaining parts of this paper, we first briefly review the previous works on clustered WSNs in which the hot spot issue is addressed and explain the major deficiencies of these past efforts in Section II. Then, our mathematical approach on how to equalize node lifetimes across the network is provided in Section III. Section IV presents the EC algorithm based on this analysis and Section V provides the details of a simple data collection protocol and the calculation of its energy consumption levels. Then, Section VI demonstrates EC’s performance results with comparison to two previous works, UCR [12] and HEED [10]. Finally, Section VII concludes the paper.

II. RELATED WORK

Clustering in WSNs is a popular technique to organize and manage the network efficiently. One major issue is to relieve CH nodes of their high load and energy consumption. LEACH [14] is a well-known clustering protocol in which the CH role is periodically rotated among nodes to achieve this balance. However, LEACH requires all CHs to perform direct transmissions to the network’s sink, thus it suffers from the cost of long-distance transmissions. As a result, the nodes that are far away from the sink drain their energy much earlier than others. To cope with this problem, EECS [15] allocates fewer number of member nodes to clusters with longer distances to the sink. Nevertheless, it is still based on single-hop transmissions to the sink from the CHs and is not scalable to large-scale networks. To avoid the high cost of long-range transmissions, HEED [10] adopts multihop inter-cluster communication and further selects its CHs based on the residual node energy levels. However, in HEED, hot spot issue appears in areas that are close to the sink, as nodes in such areas need to relay incoming traffic from other parts of the network.

To address the hot-spot issue, UCR [12], EEDUC [8], MRPUC [16] and UCS [17] propose using multihop routes to the sink and conclude that the sizes of clusters should be smaller as they approach the sink. The main idea here is to compensate for the high inter-cluster communication load by reducing the cost of intra-cluster communications. With small cluster sizes, the high load of incoming data is claimed to be distributed among more clusters, effectively reducing the load of each CH near the sink. However, this might cause too many clusters to be formed around the sink and a significant number of summary packets to be produced when approaching the sink. The result is a higher traffic load than predicted. Therefore, an analytical study is required to balance the intra-cluster and inter-cluster energy consumption amounts while considering the varying traffic load at different locations of the network.

Although a basic analysis of energy consumption for clustering is conducted in a few existing works, such as [16], [17], [18], they have some deficiencies. For example, the analysis of energy consumption in control overhead caused by route discovery and cluster formation is not fully covered. Furthermore, some key parameters are determined via complex experiments [18], which is an impractical technique.

Another issue is that clustering solutions like PEBECS [18] and UCR [12] assume network-wide announcements during the cluster formation process. However, such an assumption not only reduces energy efficiency, but also limits the applicability to small-scale networks only.

In short, there is a need for a comprehensive analysis of the total energy consumption in multihop data delivery in clustered WSNs. Such an analysis should be based on an energy-efficient data routing and clustering protocol that avoids using network-wide broadcasts and reduces control overhead. Furthermore, to establish the load balance in a WSN, this trade-off between the distance to the sink and the cluster sizes should be studied analytically but not experimentally, before setting up the network hierarchy.

III. ENERGY-EFFICIENT CLUSTERING (EC)

A. Preliminaries

In this work, we consider a multihop data collection scenario in a WSN with uniformly distributed node locations. Each sensor node makes observations, produces a single data packet, and then transmits this packet to its associated CH. Then, each CH node collects the observation packets from its associated member nodes and combines them to produce a single summary packet representing the cluster. Summary packets travel through the network’s CH-backbone towards the sink in multiple hops. This three-step process is referred to as a single data collection round (DCR) of the entire WSN operation.

1) Trade-offs: Equalization of node energy consumption levels in a multihop data collection scenario has two trade-offs: (i) There is a higher traffic load on nodes closer to the data sink in terms of hop-distance. (ii) Having clusters of large sizes produces shorter routes but increases intra-cluster communication costs. On the other hand, forming many small clusters generates affordable intra-cluster costs, yet longer multi-hop routes are generated which requires more packet transmissions, and more summary packets are generated in the network, which increases the total relayed traffic. Hence, having smaller clusters leads to a larger inter-cluster communication cost. Therefore, the analysis should take into account the hop distances to the sink node.

2) Hop distances to the sink: The hop distance to a sink node in a network area with length $X$ and width $W$, where the sink node is located at one edge, forms a wave-like propagation pattern [19] outwards from the sink. Figure 1 illustrates this pattern for a sample randomly deployed network, where
nodes at different hop distances to the sink are denoted by different symbols. The area in which nodes of a particular hop distance \( i \) reside can approximately be represented by a rectangular region \( R_i \). The widths of these regions may not be equivalent and are random variables depending on node locations and sensor communication range. However, we can denote the average region width by \( a \). We calculate \( a = 71 \) \( m \) for a node density of \( \sigma = 0.025 \) nodes/\( m^2 \), using the energy model in Section V-A (see the Appendix for details at [20]).

![Fig. 1. Hop distances to the sink and rectangular regions.](image)

3) Approximate equalization of energy levels: \( \tau(i) \): Accumulation of packets from outer regions towards the data sink creates higher traffic loads at closer locations to the sink. Since this load is distributed among the sensors of each region via rotation of the CH-role, sensors in a particular region have “approximately” equal rates of energy consumption. With this, the lifetimes of all sensors in region \( R_i \), are treated as the same, and denoted by \( \tau(i) \). This is the reason why we claim that our approach provides approximate equalization of node energy levels. Our task is now to ensure that similar energy levels are maintained at different regions throughout the lifetime of the WSN.

B. A Generic Approach to Equalization of Regional Lifetimes: EC Algorithm

1) Distribution of CH nodes in the network: \( p_i \): Under the two trade-offs in Section III-A1 that affect node energy consumption, we strive to strike the balance between a cluster’s radius and its hop distance to the sink. It is obvious that the radius of a cluster in a region \( R_i \) is related to the number and density of CH nodes in \( R_i \). This suggests that CH nodes should be distributed with different density at different hop distances to the sink. For instance, region \( R_i \) contains \( n_i \) CHs. Therefore, the probability \( p_i \) that an individual node becomes a CH in region \( R_i \) can be found by:

\[
p_i = \frac{n_i}{aW\sigma} \Rightarrow n_i = p_iaW\sigma, \tag{1}
\]

where the average number of nodes in region \( R_i \) is \( aW\sigma \). Due to the uniform distribution of node locations and energy levels, cluster areas can be approximately represented by circular sub-regions of radius \( r_i \) within region \( R_i \). Since there is a single CH inside each cluster, the probability of a sensor in region \( R_i \) to become a CH can be approximated by:

\[
p_i = \frac{1}{\pi r_i^2 \sigma} \Rightarrow r_i = \sqrt{\frac{1}{\pi \sigma p_i}}. \tag{2}
\]

2) EC Algorithm: The purpose of the EC Algorithm is to determine the probability values \( p_i \) while equalizing and reducing energy consumption levels in the network. Our specific energy equalization goal is to ensure that we have similar lifetime values at different hop distances to the sink. This means that we aim to obtain \( \tau_1 = \tau_2 = \tau_3 = \ldots = \tau_K \) for \( K \) regions.

Denoting the energy consumption in \( R_i \) within a DCR as \( E_{DCR}(R_i) \), we have \( \tau(i) = \frac{E_0aW\sigma}{E_{DCR}(R_i)} \), where \( E_0 \) is the average initial sensor energy. We would like to equalize values of \( \tau(i) \) to a value \( L \), which is as large as possible since our goal is also to extend network lifetime. Therefore, we have:

\[
\frac{E_0aW\sigma}{E_{DCR}(R_1)} = \ldots = \frac{E_0aW\sigma}{E_{DCR}(R_i)} = \ldots = \frac{E_0aW\sigma}{E_{DCR}(R_K)} = L. \tag{3}
\]

The unknowns of this problem are the individual probability values \( p_1, p_2, p_3, \ldots, p_K \) that appear in the expression for \( E_{DCR}(R_i) \). Although the energy value \( E_{DCR}(R_i) \) is dependent on which particular set of protocols is used to deliver data to the sink over multiple-hops, we can consider the worst case scenario and regard \( E_{DCR}(R_i) \) as a non-linear equation of \( p_1, p_2, p_3, \ldots, p_K \) in general. In case \( E_{DCR}(R_i) \) is a linear function, the following sequence of operations are simpler, yet we provide the general solution methodology.

Our strategy is simple: We start by assigning an initial value \( L_0 \) to the lifetime \( L \) and also set \( \tau(i) = L \) for all \( i \) in order to solve for the corresponding value of CH probability \( p_i \). Then, we update \( L \) iteratively until a valid maximum value of \( L \) is obtained. Algorithm 1 outlines this strategy. The function \( \text{calculatePs}(L) \) calculates values of \( p_1, p_2, \ldots, p_K \) for the value of \( L_{t+1} \) at iteration step \( t \). The main loop first finds the next value of \( L \), \( L_{t+1} \), using the current value \( L_t \). Then, the next probability set \( p_{k+1} \) is calculated using the function \( \text{calculatePs}(L) \), which gets \( L = L_{t+1} \) as its input. Here, the interesting module of EC is line 6 in Algorithm 1 that determines the next value of \( L_{t+1} \) given the current values of \( L_t \) and \( p_t \). This module depends on the round energy consumption in each rectangle \( i, E_{DCR}(R_i) \), and hence the individual data routing protocol used to deliver the packets to the sink. This constitutes the module that needs to be filled in as a separate add-on to EC for a particular protocol.

C. Application of EC to a simple and energy-efficient data collection protocol

In this section, we apply the EC algorithm to a simple data collection protocol explained in detail in Section IV and find the probability values of nodes to be selected as a CH in each region, \( p_K, p_{K-1}, \ldots, p_{t+1} \). Such information tells us the number and the density of CHs and hence the cluster sizes corresponding to each hop distance to the sink.

Note that the details of the data collection protocol are irrelevant at this point as we are only interested in how to use its resulting energy expression \( E_{DCR}(R_i) \) in EC. In order not to interrupt the logical sequence of ideas of the article, we defer the details of this protocol to Section IV. \( E_{DCR}(R_i) \) is given by Equation 17, which is represented here as a function
Algorithm 1 EC Algorithm
Ensure: $\tau(K) \approx \ldots \approx \tau(i) \approx \ldots \approx \tau(1) \approx L$
1: $t \leftarrow 0$;
2: $P_t = P_0 = \{p_0, p_0, \ldots, p_0\}$;
3: $L_{t+1} \leftarrow L_0$;
4: $P_{t+1} = \{p_1, p_2, \ldots, p_K\} \leftarrow \text{calculatePs}(L_t)$;
5: while $P_{t+1} = \{p_1, p_2, \ldots, p_K\}$ are Real and Non-negative do
6: Determine $L_{t+1}$;
7: $P_{t+1} = \{p_1, p_2, \ldots, p_K\} \leftarrow \text{calculatePs}(L_{t+1})$;
8: $P_t \leftarrow P_{t+1}$;
9: $L_t \leftarrow L_{t+1}$;
10: % An exit condition that meets a certain requirement specific to the protocol
11: if $C(L_{t+1}) = \text{true}$ then
12: return $P_{t+1}, L_{t+1}$
13: end if
14: $t \leftarrow t + 1$;
15: end while
16: return $P_t, L_t$;

\textbf{calculatePs}(L):
1: Solve $\tau(K) = L$ for $p_K$;
2: Solve $\tau(K - 1) = L$ with $p_K$ for $p_{K-1}$;
3: ;
4: Solve $\tau(1) = L$ with $p_K, p_{K-1}, \ldots, p_1$ for $p_1$;
5: return $p_1, p_2, \ldots, p_K$;

$f(p_1, \ldots, p_K)$, yielding:
$$\tau(i) = \frac{E_0 a W \sigma}{f(p_1, \ldots, p_K)}.$$ \hfill (4)

1) Step 1: Solving for $p_i$ values: calculatePs(L): There is a property of the lifetime equations $\tau(i) = L$ that we can exploit: Since the $K^{th}$ region is the outermost region and does not relay any traffic from other regions, $p_K$ is independent from $p_{K-1}, \ldots, p_1$. Therefore, for a given value of $L$, $\tau(K) = L$ can be solved for $p_K$ on its own. Then, the solution for $p_K$ can be used in the next equation $\tau(K-1) = L$ to determine $p_{K-1}$, and so on. Therefore, each equation $\tau(i) = L$ has a single unknown $p_i$ since $p_K, p_{K-1}, \ldots, p_{i+1}$ are already calculated, for a given value of lifetime $L$. Note that this is true for all data routing protocols in data collection scenarios towards a single network sink.

The protocol we use in Section IV yields Equation 17 for the round energy that turns into a polynomial equation of $p_1$ when $p_K, p_{K-1}, \ldots, p_{i+1}$ are constant. By re-organizing $\tau(i) = L$ as a second order polynomial $Ap_i^2 + BP_i + C = 0$, we can find the coefficients of this polynomial as:

$$A = \alpha \epsilon_0 (W^2 + 4a^2)^2 + \alpha \sum_{i<j\leq K} (Wa\sigma e_\psi p_j) + (\epsilon_0 D^4 - \epsilon_r),$$
$$B = \sum_{i<j\leq K} (\epsilon_2 (2a^2 - \frac{4}{9} + (a^2 + 3 - T)e_r + (T + 3)\epsilon_r)),$$
$$+ \sum_{i<j\leq K} (2a^2 W + 4a^2) + \sum_{i<j\leq K} (p_j - \frac{B}{E}),$$
$$+ \sum_{i<j\leq K} (e_r + e_t + E_{\text{hit}} - \frac{4f_\psi}{36\pi\sigma} + (e_r + e_t + \epsilon_0 D^4) \sum_{i<j\leq K} p_j),$$
$$C = \sum_{i<j\leq K} (T^2 e_r + \frac{f_\psi}{9\pi\sigma} (T + \frac{4}{9})) + \frac{4f_\psi}{9\pi\sigma}. \hfill (5)$$

Once again, the derivation of the expressions in Equation 5 is purely mathematical, not related with the focus of the paper, and provided here for the sake of completeness. Interested readers can use the information in Section IV to derive these expressions.

Equation 5 provides us with the calculation method of individual $p_i$ values. Note that this is the duty of the function calculatePs(L) in Algorithm 1. Each line in calculatePs(L) uses Equation 5 to solve for a $p_i$, starting from the outermost region $i = K$.

2) Step 2: How to iterate $L$: Determine $L_{t+1}$: Here, we determine how the data collection protocol we use in Section IV makes the iterations of lifetime $L$, i.e. calculation of $L_{t+1}$ given $L_t$ and $P_i$. Hence, the following analysis is specific to that protocol. A similar analysis has to be followed for any other data collection protocol in order to determine an iteration policy for $L$.

In Equation 5, $\sqrt{B^2 - 4AC} \geq 0$ must hold so that the roots of the equation are not imaginary. Since $A, C > 0$ and the roots should also be positive values in $[0, 1]$ (as $p_i$s are probability values), then $B < 0$ must hold. Therefore, we have:
$$B \leq -2\sqrt{AC}. \hfill (6)$$

We first simplify Equation 5 as a function of the number of nodes $n$. Considering the number of CH nodes that forward traffic to region $R_i$ given by $\sum_{i<j\leq K} Wa(p_j \approx \psi/n)$, to be at the same order as the number of nodes $n$, and representing node density $\sigma = \frac{n}{WX}$, this gives:

$$A = C_1 + C_2 + \psi/n^2, \quad B = \frac{C_{\psi}}{n} + C_4\psi/n - \frac{E_0}{T}, \quad C = C_5 + \frac{C_6}{n}, \hfill (7)$$

where $C_1$ to $C_6$ are constants. By applying Equation 7 to Equation 6, we get $L \leq L_{\text{max}}(n)$, given by:

$$L_{\text{max}}(n) = \frac{E_0}{C_4\psi/n + 2\sqrt{C_2C_3\psi/n^2 + C_2C_6\psi/n + C_1C_5 + \frac{C_6}{n}}}. \hfill (8)$$

$L_{\text{max}}(n)$ is an upper bound on the range of lifetime values $L$ as a function of the total number of nodes $n$. The solutions beyond the limit imposed by Equation 8 have imaginary values. Hence, we seek this bound $L_{\text{max}}(n)$ by greedily increasing $L$ until we start to get imaginary solutions and stop at the largest non-imaginary set of solutions. \footnote{An imaginary solution set means that the lifetime cannot be equalized among different regions for the current value of $L$.} Therefore, starting from $L = 1$ and incrementing $L$ by 1 until $L_{\text{max}}(n)$, the number of iterations necessary for EC to find $L_{\text{max}}(n)$ is given by $L_{\text{max}}(n)$ itself.

The complexity of the EC algorithm is bounded by $L_{\text{max}}(n)$, which does not exponentially grow as we increase $n$. In fact, the denominator goes to infinity when $n$ goes to infinity and when $n = 0$. In between the two, the denominator has a minimum value. Hence, with increasing $n$, $L_{\text{max}}(n)$ first increases, arrives at its maximum at a point, and then gradually decreases. This is attributed to the fact that addition of more
nodes leads to density increase in the network. Having more nodes may provide additional opportunities to nodes to find closer nodes to forward their data towards the sink, which decreases transmission energy. However, addition of even more nodes does not help much to decrease this consumption, but rather increases the overhead incurred by packet receptions and additional data collection.

3) **Sample results of EC**: The results of Algorithm 1 for the protocol in Section IV are presented in Figure 2 for various node density settings. Here, the probability of a sensor node to be selected as a CH in its region is shown with respect to the hop distance to the data sink. As can be observed, it is more likely for a node to play the role of a CH when that node is closer to the data sink. This is inline with UCR [12] that has a similar data routing protocol. However, UCR essentially presumes such type of CH distribution.

![Figure 2](image)

**Fig. 2.** Output $p_j$ values in Algorithm 1 for the protocol in Section IV.

**IV. A MULTIHOP DATA COLLECTION PROTOCOL FOR WSNs**

In this section, an energy-efficient multihop data routing solution for WSNs organized as clusters is briefly outlined. There are three reasons for presenting such a protocol:

1) To complement EC’s energy equalization and conservation features with a protocol that also targets at energy-efficiency and reduces its overall energy consumption level via using less control messages.

2) To make comparisons of EC with existing clustering solutions that target at energy efficiency in multihop data delivery.

3) To understand whether EC actually achieves energy-efficiency and equalization with its output probability values for CH-selection.

**A. Cluster head selection**

Cluster formation is performed as a distributed algorithm at the beginning of each data collection round, DCR. This involves election of CH nodes among a set of candidates followed by node-CH associations.

1) **Selection of CH-candidates**: To determine the CH candidates, a probability scale is assigned to each sensor. According to this value, each sensor decides on becoming a CH-candidate. Basically, the probability to become a CH-candidate, $T$, is scaled by the ratio of initial sensor energy level to the average initial energy of the network, $E_0$. For a node $j$ in region $R_i$, the resulting probability becomes $P(j) = T \cdot \frac{E_j}{E_0}$. Computation of $P(j)$ is performed only once right after network initialization.

At the beginning of each DCR, each node $j$ picks a random number on $[0, 1]$. If the number is less than $P(j)$, then the node is a CH-candidate. With this mechanism, approximately a ratio $T$ of all nodes are elected as CH-candidates. In simulations, we pick $T = 10\%$ as in [12].

**Discussion: The candidate selection probability $P(j)$**: The selection of $P(j)$ would be more up-to-date if the residual node energy levels $E_j$ are considered instead of the initial energy levels $E_0$, hence $P(j) = \frac{T \cdot E_j}{E_0}$, where $E_j$ is the average residual energy level within a region. However, this would require each node to notify all others in its region of its energy value, which could only be achieved by region-wide broadcasts; a quite high message overhead. Alternatively, a central node in each region could gather the energy levels and then distribute the average value to all sensors in the region, which is a slightly better scheme. Nevertheless, this would add the additional complexity of choosing and replacing such a central node. Another method would be the use of counters at each node to keep track of the number of times they take the CH role. However, this also requires later negotiations among nodes.

To avoid all these issues, we use the initial energy levels for selecting the CH-candidate nodes. Since the frequency of being selected as a CH-candidate is proportional to the initial energy levels and the CHs are eventually selected among these candidates, the resulting frequency of having the CH-role and the corresponding energy consumption are on the average approximately proportional to the initial energy levels. Therefore, this choice is a reasonable method towards balancing energy consumption levels while preventing additional overhead. Note that node residual energy levels are taken into account during the selection of the actual CHs, as explained next.

2) **Selection of CHs from CH-candidates**: Upon being selected, each CH-candidate in region $R_i$ transmits a “CH-announcement” packet and advertises its residual energy level within a neighborhood of radius $r_i$. $r_i$ is determined by the EC algorithm and is related to $p_i$ by Equation 2.

A CH-candidate monitors advertisements from other candidates and defers from acting as a CH if a higher energy level is reported by another. Eventually, the candidates with the highest residual energy among their neighboring CH-candidates become the CHs during that particular DCR. (If a CH-candidate receives no advertisement packets for a period of $T_{wait}$, it automatically becomes a CH node.) This mechanism enables the choice of the actual CH nodes to be based on the most recent sensor energy stocks. The pseudocode of the algorithm for CH selection is available at [20].

**B. Cluster formation**

After the CHs are elected, each CH transmits a “CH-announcement” packet within an area of transmission radius $\alpha r_1$ and informs other sensors of its availability as a CH. This CH-announcement range is a multiple of $r_1$, $\alpha \sqrt{\frac{1}{p_i} p_i} = \alpha r_1$, selected to ensure that each non-CH node receives at least one announcement packet and can associate to a CH. To
ensure reception of announcement packets by other nodes, a straightforward method is to send region-wide broadcasts. However, this potentially causes high transmission energy cost; a fine tuned value is required. Thus, $\alpha$ is a system parameter tuned to achieve high CH-association probability for non-CH nodes while avoiding an unnecessarily large transmission range.

Considering a uniform distribution of CH nodes in each region, the number of nodes in a given area has a Poisson distribution [19], [21]. Hence, the probability that a non-CH node has at least one CH neighbour within a circular area of radius $\alpha R_1$ in region $R_i$ is $1 - e^{-\sigma p_i \pi \alpha^2 R_1^2}$. To ensure a high rate of CH-association, we seek at least 99% average connectivity probability, hence $1 - e^{-\sigma p_i \pi (\alpha R_1)^2} \geq 0.99$. This leads to $\sqrt{\frac{2ln10}{\sigma p_i}} = \alpha R_1$, yielding $\alpha \geq \sqrt{2ln10}$. Hence, we select $\alpha = \sqrt{2ln10}$.

Each sensor may collect announcement packets from multiple CHs and selects the CH that has generated the announcement packet with the highest RSSI as the ideal CH to associate to. Nodes associate to CHs via sending a “CH-association” request and upon reception of a subsequent “CH-confirmation”.

At the end of the cluster formation phase, there may still be a few sensors that have not joined any clusters as they may not have received any announcement packets. To recover from such cases, a sensor with no CH-association gradually increases its transmission range and seeks the closest CH to associate.

The pseudocode of the algorithm for cluster formation is available at [20].

C. Message complexity of Clustering

In a WSN of $N$ nodes, $NT$ nodes advertise as CH-candidates, producing a total of $NT$ messages. Eventually, $M$ CH nodes are selected, which then announce their role as a CH with a total of $M$ CH-announcement messages. Sensor nodes choose a CH to join and send CH-association requests, incurring an additional cost of $N - M$. For each request, a CH-confirmation message is generated. As a result, the total message complexity in cluster formation is approximately $NT + M + 2(N - M) = (2 + T)N - M = O(N)$.

D. Distributed inter-cluster routing

The routing algorithm is based on two ideas: (1) Reducing the overhead in route discovery, (2)Balancing energy consumption among all CHs. To achieve these goals, a simple scheme is used: Basically, a CH node in region $R_i$ chooses its next hop towards the sink in the neighbor rectangular region, $R_{i-1}$. The CH transmits a route request packet with a range of $\sqrt{W^2 + 4d^2}$, sufficiently large to cover $R_{i-1}$. Each receiving CH in $R_{i-1}$ generates a reply packet and starts a route reply timer with an expiration time inversely proportional to its residual energy level. The first node that has an expired timer actually makes the transmission of a route reply packet back to the requester CH in $R_i$, while the rest quietly cancel their timers upon hearing this reply. This guarantees that a single reply packet is sent and thus prevents excessive message overhead. Furthermore, by considering the residual energy levels, priority is given to nodes with higher resources; a policy towards balancing energy consumption in the entire network.

V. ENERGY CONSUMPTION CALCULATIONS

In this section, we provide the value of energy consumption $E_{DCR}(R_i)$ of a region $R_i$ during a DCR period. Recall that this value is needed by the EC algorithm in order to compute the equalized lifetime value and node probability values of becoming a CH. For the sake of clarity, the text in this section may briefly mention what is already explained in detail in Section IV.

Denoting the total energy consumption in $R_i$ during a DCR as the round-energy, $E_{DCR}(i)$, and the total initial deployment energy in $R_i$ as $E_0(R_i) = \sum_{j \in R_i} E_0(j) = \sum_{j \in R_i} \bar{E}_0(1 + x_j) = \bar{E}_0 a W \sigma$, where $\bar{E}_0$ is the average deployment energy and $x_j$ denotes the variation of energy at node $j$. The lifetime of $R_i$ is approximately $\tau(i) \approx \frac{\bar{E}_0(R_i)}{E_{DCR}(i)}$. The round energy is the sum of energy consumption values for cluster formation, route discovery, and data communication events, given by $E_{DCR}(i) = E_{Cluster}(i) + E_{Route}(i) + E_{Comm}(i)$.

In the following, individual parts of $E_{DCR}(i)$ are separately calculated. Throughout the analysis, a random variable is depicted in bold-face fonts.

A. Sensor energy consumption model

EC is independent of the particular sensor energy model, yet considers a popular one mentioned in previous works [3], [13], [14], [22], given by:

$$E = \begin{cases} l(e_t + e_r + \epsilon_j d^2) & \text{if } d < d_{th} \\ l(e_t + e_r + \epsilon_{mp} d^4) & \text{if } d \geq d_{th} \end{cases} \quad (9)$$

In Equation 9, $E$ is the total energy dissipated to deliver a single $l$-bit packet from a transmitter to its receiver over a single link of distance $d$. The baseline energy consumption levels at the transmitter and receiver radios are indicated by $e_t$ and $e_r$, respectively. The transmission energy consumption is denoted by either $\epsilon_j d^2$ or $\epsilon_{mp} d^4$, depending on the distance $d$ of the link between the two nodes and a distance threshold, $d_{th}$. For $d \leq d_{th}$, $\epsilon_j$ is used to reflect “free-space” conditions, while $\epsilon_{mp}$ represents longer links potentially affected by “multi-path” fading [15].

B. Energy consumption in Cluster Formation

The selection of CHs is a two-stage process as explained in detail in Section IV-A. Designating the length of a control packet as $l_o$, we obtain the total clustering energy consumption during a DCR in $R_i$ as:

$$E_{Cluster}(i) = Wa \sigma T \left[ l_o (e_t + \frac{\epsilon_j}{\pi \sigma p_i}) + \frac{T}{p_i} - 1 \right] l_o e_r$$
$$+ Wa \sigma (1 - p_i) l_o \left( e_t + \frac{\epsilon_j \sigma^2}{9 \pi \sigma p_i} + e_r \right)$$
$$+ Wa \sigma p_i l_o \left( e_t + \frac{\epsilon_j \sigma^2}{\pi \sigma p_i} + \pi e_r \frac{\sigma^2}{\pi \sigma p_i} - 1 \right)$$
$$+ Wa \sigma p_i l_o \left( e_t + \frac{\epsilon_j \sigma^2}{\pi \sigma p_i} \right) + Wa \sigma (1 - p_i) l_o e_r. \quad (10)$$
This equation can be briefly explained as follows. A ratio $T$ of all nodes initially are CH-candidates inside each region $R_i$, hence $W \sigma T$ candidates per region. The candidate nodes compete with each other to become a CH and announce their candidacy within a competition range of radius $r_i$. These candidate announcements are received by peer nodes in each region, which is on the average $\pi r_i^2 \sigma T - 1$ sensors per candidate. The first term in Equation 10 is for the candidate nodes to announce their candidacy and for the reception of these announcements by peer nodes in the region. Upon being selected, each CH announces its role with a CH-announcement packet that is received only by the nodes inside its announcement range. The second term stands for these events. Each non-CH node needs to send a control packet to associate with a CH that then replies back with an association message, which forms the third term. Finally, the last two terms are for all CHs in region $R_i$ to distribute their time schedules among cluster nodes.

C. Energy consumption in Route Discovery

Route discovery in EC is achieved by a reactive routing mechanism, explained in detail in Section IV-D. Basically, a CH requesting a route to the sink transmits a route request to its neighboring CHs closer to the sink in terms of hop distance. The range of this transmission is equivalent to the diagonal distance of two adjacent rectangular regions (see Figure 1). This is equal to $\sqrt{W^2 + (2a)^2}$, a sufficiently large range to cover all nodes in a neighbor region. Considering all requests emerging from outer regions as a load on region $R_i$, the number of such request can be found by $\sum_{i<j<k} (W a \sigma p_j)$. Equation 11 provides the total energy consumption per round in $R_i$ due to route discovery message exchanges.

The first term designates the reception of routing requests coming from outer regions to $R_i$ by all CHs in $R_i$, plus the following acknowledgement packet sent by a single CH in $R_i$ to the outer region. The second term stands for the further route discovery initiated by $R_i$ towards the sink.

$$E_{Route}(i) = \sum_{i<j} (W a \sigma p_j) + l_o \left[ W a \sigma t_{eT} + e_t + \epsilon_{mp}(W^2 + 4a^2)^2 \right]$$

$$+ l_o \left[ e_r + e_t + \epsilon_{mp}(W^2 + 4a^2)^2 \right] \sum_{i<j<k} (W a \sigma p_j).$$

D. Energy consumption in Data Communication

Data communication events consist of intra-cluster communications, inter-cluster communications, and data processing. Hence, the total communication energy consumption in a round is $E_{Comm} = E_{Intra} + E_{Inter} + E_{Proc}$.

1) Inter-cluster communication energy consumption: Each region $R_i$ contains $n_i$ clusters, generating $n_i$ summary packets in total. These packets are then forwarded to the next region. In addition to its own packets, region $R_i$ relays incoming packets from outer regions. Hence, the total number of packets transmitted by region $R_i$, $n_T(i)$, is approximated by $n_T(i) = \sum n_j$, for $i \leq j \leq K$. Following the energy consumption model in Equation 9, this is equivalent to a transmission energy cost of $E_{T_r}(i)$ during a DCR, given by $E_{T_r}(i) = \sum_{i} (e_t + \epsilon_{mp}D^4)t$, where $l$ is the average packet length. The CH nodes in $R_i$ also consume sensor energy while receiving these incoming packets from outer regions. This is equal to $E_{Recv}(i) \approx e_r(n_T(i) - n_j)l$. Hence, we approximate the total inter-cluster communication cost for $R_i$ during a DCR as:

$$E_{Inter}(i) \approx l(e_r + e_t + \epsilon_{mp}D^4)n_T(i) - le_r n_i.$$

2) Intra-cluster communication energy consumption: During a DCR, each sensor encapsulates its observed information in a data packet and then transmits this to the corresponding CH. The CH accumulates all observation packets and combines them in a single summary packet to summarize the observation of the area within the cluster boundaries. In region $R_i$, there are approximately $a W \sigma$ nodes. Therefore, with $n_i$ CHs, the total number of observation packets delivered to the CHs in $R_i$ is $a W \sigma - n_i$.

Similar to the calculation in Section V-D1, we consider both transmission and reception events and approximate the total intra-cluster energy consumption, $E_{Intra}(i)$, in region $R_i$ as:

$$E_{Intra}(i) = l \left( \sum_{i \leq j < k} (e_t + \epsilon_{fs}d_j^2) \right) + l(W a \sigma - n_i)e_r,$$

where $d_j$ is the distance between node $j$ in $R_i$ to its associated CH.

3) Data processing energy consumption at a CH: Although comparably minimal, data processing for summarizing packets at each CH consumes some sensor energy. This is linearly proportional to the number of CHs. Designating $E_{bit}$, [3], [13], [14], [22], as the energy necessary to process one bit of data, we have:

$$E_{Proc}(i) = l a W \sigma E_{bit}.$$  

4) Energy consumption in data communications per round: Using Equations 12, 13, 14, we derive the approximation for the total energy consumption in data communications during a single round of data collection in region $R_i$ as:

$$E_{Comm}(i) = l \left( \sum_{i \leq j < k} (e_t + \epsilon_{fs}d_j^2) \right) + l(W a \sigma - n_i)e_r + l a W \sigma E_{bit} + l(e_r + e_t + \epsilon_{mp}D^4)n_T(i) - le_r n_i.$$  

Due to the uniformly distributed node locations, using Equation 2, the average distance $d$ between a sensor and its CH, $d_{avg}$, can be calculated as $d_{avg} = \frac{1}{2}r_i$. Here, the probability that a sensor is located at a distance $d$ to its CH in the circular cluster area is given by $\frac{2r_i d}{\pi r_i^2}$. Replacing $n_i$ by Equation 1 and $r_i$ by Equation 2, we get:
\[
E_{\text{Comm}}(i) = laW_\sigma \left( e_r + e_t + E_{\text{Bit}} - \frac{4e_{fs}}{9\pi \sigma} \right) + laW_\sigma \left( e_r + e_t + \frac{4e_{fs}}{9\pi \sigma} p_i \right) + laW_\sigma (e_r + e_t + \epsilon_m D^4) \sum_{i \leq j \leq K} p_j. \tag{16}
\]

E. Total Energy Consumption per Round in a Region

We can finally combine all expressions for energy consumption, Equations 10, 11, and 16 to obtain the total round energy consumption \( E_{\text{DCR}}(i) = E_{\text{Comm}}(i) + E_{\text{Cluster}}(i) + E_{\text{Route}}(i) \) as a function of the probability values, \( p_1, \ldots, p_K \):

\[
f(p_1, \ldots, p_K) = \left. \begin{array}{l}
laW_\sigma (e_r + e_t + E_{\text{Bit}} - \frac{4e_{fs}}{9\pi \sigma} - (e_r + 2e_r) p_i - \frac{4e_{fs}}{9\pi \sigma} \frac{1}{p_i}) \\
+ laW_\sigma (e_r + e_t + \epsilon_m D^4) \sum_{i \leq j \leq K} p_j \\
+ WaT \left[ p_i (e_{r} - 3e_{r}) + \frac{1}{p_i} (2T_{e_{r}} - \frac{1}{p_i} (T + \frac{1}{p_i})) \right] \\
+ WaT \left[ (\alpha^2 + 2 - T)e_{r} + (T + 1)e_{t} + \frac{1}{p_i} (2\alpha^2 - \frac{1}{p_i}) \right] \\
+ T_{s} \left[ \epsilon_m (W^2 + 4a^2) + 2e_{t} + e_{r} \right] \sum_{i < j \leq K} (WaT p_j) \\
+ p_{i} T_{s} \left[ e_{r} + e_{t} + \epsilon_m (W^2 + 4a^2) + \sum_{i < j \leq K} W^2 a^2 \sigma^2 p_j \right] \right].
\]

VI. PERFORMANCE EVALUATION OF EC

In this section, the performance of EC is compared with HEED [10] and UCR [12]. HEED is a distributed clustering algorithm, where CHs relay data to a sink node via multihop routing. It has an iterative CH selection mechanism in which the probability for each node to become a CH depends on its residual energy. When a CH candidate is not selected as a CH node in a round, it doubles its probability of selection so that it will have a better chance in the next round. Although widely accepted as a major clustering algorithm, HEED does not address the hot-spots of the network, hence no lifetime equalization mechanism is involved.

Different from HEED, UCR also additionally targets at equalizing sensor lifetimes via CH competitions. UCR has a similar CH-competition mechanism with EC. However, the CH competition range of UCR is simply determined by an intuitive calculation based on the distance to the sink; the closer a node is to the sink, the more likely it will be selected as a CH. In addition, UCR assumes a network-wide broadcasting, which wastes energy on unnecessary transmissions; a drawback when compared to the other two algorithms.

A. Simulation parameters

Our simulations are performed using MATLAB. 1000 nodes are uniformly distributed in a network area of 100m \( \times \) 400m, leading to a node density of \( \sigma = 0.025 \) nodes/m\(^2\). This initial set of simulations demonstrate the performance of EC for a fixed node density. We also evaluate the effect of node density on EC, varied between 0.0065 and 0.05 nodes/m\(^2\).

Similar to the current literature, we deploy the sink node slightly outside of an edge of the network area, at coordinates (450m, 50m). Nodes have initial unequal energy levels in the range [2, 4]J. The average initial node energy level \( E_0 = 3 \) as a manufacturer value is known by all nodes. The parameters for the node-energy model are as follows: \( \epsilon_{mp} = 0.0013pJ/\text{bit/m}^3 \), \( \epsilon_{fs} = 10pJ/\text{bit/m}^2 \), \( e_r = e_r = 50nJ \), \( d_{Th} = 87m \), \( E_{\text{Bit}} = 5nJ/\text{bit/signal} \), data packets size: 4000 bits, and control packet size: 200 bits.

B. Output of EC Algorithm

Prior to simulations, we first run the EC algorithm to determine the competition ranges of nodes in different regions (\( r_j \) values) that provides a valid solution to the joint problem of network lifetime extension and node energy equalization (see Figure 2).

C. Stable Operation Period (SOP)

The number of “alive” nodes over simulation time is illustrated in Figure 3(a). We define Stable Operation Period (SOP) as the period of time until the first node in the network depletes its energy. The figure shows that the SOP of EC is around 4500 rounds, whereas the SOPs of HEED and UCR are around 1250 and 1500 rounds, respectively, hence EC has an overwhelming performance in extending SOP. Our simulations on different node density settings, \( \sigma = 0.00625 \), \( \sigma = 0.0125 \), and \( \sigma = 0.05 \) nodes/m\(^2\), also demonstrate similar results. These results are omitted due to the limit on figures and number of pages, yet they are available online at [20].

Another observation is the shape of the graphs. The sharp decrease at UCR and EC in Figure 3(a) demonstrates that energy equalization is achieved until the SOP instance. When SOP is reached, a large number of nodes start to deplete energy resources, leading to this quick decline. On the other hand, HEED shows a gradual decrease in the number of alive nodes, which is due to the fact that energy equalization is not coordinated and some nodes run out of battery energy much more quickly than others. This is observed in all node density settings.

(a) EC improves the network’s stability. (b) EC delivers more messages to the sink until the last node is offline. [Fig. 3] SOP and total number of delivered messages.

A noticeable difference between HEED and EC is that when EC completely loses all of its nodes, HEED still has some...
nodes operational, yet with low residual energy. This raises the question as to which protocol is “better” to serve the data collection scenario. To answer this, we also compute the total number of messages that each protocol delivers during the time period from time 0 until the time instance when no more nodes are alive. Figure 3(b) illustrates the results. EC has the highest delivery amount at different node density settings. Despite the fact that HEED still has some alive nodes after EC loses its last node, the remaining set of nodes in HEED during that interval has only less than 20% of the initial number of nodes and cannot provide full network coverage.

D. Energy equalization

Figure 4 shows the residual energy levels of sensor nodes at the SOP of UCR (1500 rounds). Dots indicate sensor nodes and a larger dot is used for a higher residual energy level. As can be clearly observed, both UCR and EC achieve energy equalization with EC showing a better performance in energy conservation, whereas sensors in HEED have largely varying levels of energy. In Figure 5(b), the coefficient of variation of node residual energy levels is plotted with respect to simulation time. EC shows minimal variation in energy levels, while HEED and UCR have larger fluctuations. Furthermore, after 1500 rounds, EC nodes have significantly higher residual energy stocks compared to sensors in UCR. We can also observe an early jump for UCR in Figure 5(b) around its SOP instance. A jump is seen at the instance when many sensors have depleted their energy resources.

E. Energy conservation

Although both EC and UCR achieve energy equalization, EC outperforms UCR in energy efficiency due to its suitable choice of cluster sizes and the energy conserving features of its cluster formation mechanism. This is also observed in Figure 5(a), where the average of sensor residual energy levels are shown. Similar results are observed for different node density settings, which are available at [20].

F. Effect of node density on network lifetime

In Figure 6(a), we plot the network lifetime (SOP) with respect to different node density settings. The SOP is the highest for EC, showing a superior performance to UCR and HEED. SOP seems to “stabilize” with increasing node density with a diminishing incline. With higher node density, multihop routes in simulations can detect more direct CH routes towards the sink with potentially shorter link distances, which reduces energy consumption. After a certain density, data delivery energy consumption between cluster members and the CH increases. Furthermore, cluster formation and route discovery have a higher total overhead. Due to these reasons, SOP is slightly reduced at higher density settings.

G. Effect of network width on lifetime

Figure 6(b) illustrates the SOP of EC when we increase the network width W. The node density is also varied. Hence, the
figure shows the combined effect of node density and $W$ on SOP. The first observation is that SOP is higher for higher node density values. With additional nodes deployed in the network, the CH role can be shared among a larger number of nodes. Therefore, the network lifetime is extended. However, there is a limit as to how much a higher node density can improve the SOP. As it can be observed, the two graphs for the higher density values, $\sigma = 0.025$ and $\sigma = 0.05$ are quite close.

Secondly, when the network is sufficiently large, SOP decreases with increasing width $W$. With a larger $W$, the transmission range of inter-cluster route discovery (see Section V-C) and the average distance of inter-cluster data transmission also increase. This leads to higher energy consumption and less SOP.

Finally, for the lowest density, we observe a diminishing increase in SOP with respect to increasing $W$. This increase in a low node density setting is due to the following fact: with a small number of nodes in each region, the traffic often goes through a few nodes that are frequently selected as CHs. Hence, these nodes quickly deplete their energy. On the other hand, when we increase $W$, the inter-cluster routing algorithm can find more opportunities to select different CHs, which improves the SOP.

### H. Effect of network length on lifetime

As the network has more regions (while keeping the width of each region constant), the length of the topology is bigger. This shows the scalability of EC, UCR, and HEED with respect to network size. In Figure 6(c), this effect is illustrated for different number of regions, plotting the corresponding SOP values. We use $\sigma = 0.025$ as the node density. Out of 5 settings, EC is observed to have a consistently longer lifetime compared to UCR and HEED. As expected, lifetime decreases for all three algorithms since the network needs to forward more data due to increasing load on each region, especially at locations closer to the sink, similar to the effect mentioned in Section VI-G. However, EC demonstrates a more stable SOP performance as compared to HEED and UCR, suggesting better scalability to network size.

### VII. CONCLUSION

This work presents an energy equalization and conservation algorithm EC, suitable for multihop data delivery scenarios in clustered WSNs. EC determines the density of CH nodes in the network based on the hop distance to the network’s data sink. A simple energy-efficient data delivery protocol is proposed as a means to improve energy-conservation in multihop data delivery scenarios. This protocol is used to evaluate the effectiveness of EC’s findings on CH distribution in the network. Performance results demonstrate that EC extends network lifetime and provides equalization of node energy levels in locations at different hop distances to the sink, despite the various traffic loads at those locations. EC outperforms well-known and popular clustering algorithms HEED and UCR, which also focus on multihop data delivery in WSNs, in energy conservation and equalization.

### ACKNOWLEDGMENT

This work has been partially supported by the European Union FP7 research projects SENSEI (IST-7-215923), and EXALTED (ICT-5-258512). The views expressed are those of the authors and do not necessarily represent the projects.

### REFERENCES


Dali Wei received BSc and MSc degrees from University of Electronic Science and Technology of China in 2000 and 2003 respectively. He received PhD degree from University of Cape Town, South Africa in December 2007. He joined the Centre for Communication Systems Research (CCSR), University of Surrey, UK in March 2008. Dali Wei is coauthor of many research publications. He has served as a TPC member or reviewer for many flagship conferences or journals. His current research interests are in wireless sensor networks, mobile ad hoc networks, mesh networks and Internet of Things. He is a member of the IEEE.

Yichao Jin received his BS degree in automation from Zhejiang University, China, in 2004, and his MSc degree in mobile communication systems from the University of Surrey, UK, in 2008. Currently, he is a PhD student in the Center for Communication Systems Research, University of Surrey, UK. His research interests include short range wireless communications, network resource allocation and optimization, energy efficient clustering and routing in WSNs. He was visiting researcher at the University of Melbourne, Australia in 2009.

Serdar Vural received his MS and PhD degrees in Electrical and Computer Engineering in 2005 and 2007, respectively, from The Ohio State University, Columbus, Ohio, USA. His BS Degree is in Electrical and Electronics Engineering in 2003 from Bogazici University, Istanbul, Turkey. After completion of his PhD program, Dr. Vural worked as a Post Doctoral Researcher at University of California, Riverside from June 2008 to November 2009. He joined the Centre for Communication Systems Research in University of Surrey, Guildford, UK in January 2010. He has served as TPC member of IEEE ICDCS and PIMRC conferences, and technical reviewer for numerous archived journals and conference proceedings. Dr. Vural’s research interests are in Wireless Sensor Networks and Wireless Ad Hoc Networks, specifically in routing and medium access control protocols and systems, Wireless Mesh Networks, Wireless Metropolitan Networks, and network forensics. He is a member of the IEEE.

Klaus Moessner is a member of the Centre for Communication Systems Research at the University of Surrey, UK. Klaus earned his Dipl-Ing (FH) at The University of Applied Sciences in Offenburg, Germany, an MSc from Brunel University, UK and his PhD from the University of Surrey (UK). His research interests include reconfigurability of the different system levels, including reconfiguration management, service platforms, service oriented architectures and IMS, as well as scheduling in wireless networks and adaptability of multimodal user interfaces. He is involved in investigation and teaching of mobile service platforms, service oriented architectures, mobile service delivery and service enablers.

Rahim Tafazolli is the Director of the Centre for Communications Systems Research (CCSR), Faculty of Engineering and Physical Sciences, The University of Surrey in the UK, and the chairman of EU e.Mobility Technology Platform Expert Group. He has published more than 500 research papers in refereed journals, international conferences and as invited speaker and he is the editor of two books on “Technologies for Wireless Future” published by Wileys, Vol.1 in 2004 and Vol.2 2006. He is also the founder and the past Chairman of IET International Conference on “3G Mobile Technologies”. Prof. Tafazolli is the advisor and the consultant to a number of mobile companies and currently a member of Scientific Advisory Board of Telefonica I+D.