Self-Organized Data-Energy-Aware Clustering and Routing for Wireless Sensor Networks

Ehssan Sakhaee, Naoki Wakamiya, Masayuki Murata
Graduate School of Information Science and Technology
Osaka University, Osaka, Japan
Email: {sakhaee, wakamiya, murata}@ist.osaka-u.ac.jp

***Abstract*—In this paper we propose a clustering and routing scheme for wireless sensor networks based on a self-organizing approach. The aim of this approach is for nodes to perform an integrated emergent task (data gathering and reporting to the sink) by simply following rules based on their individual local environment. Clusterhead election is also performed in a distributed manner and is based on sensorfs relative residual energy and the relative amount of information that they need to send to the sink. Hence nodes are assumed to possess variable data sizes across the network. In the scheme clusters are formed, and decide on their roles in the data gathering and routing procedure. Clusters at the edge of the network identify themselves and become in charge of initiating the routing of data, whilst intermediate clusters await data from higher clusters from the edge of the network for further aggregation and forwarding towards the sink. Simulation results show that the clustering scheme is able to reduce the total amount of energy used up by the network and efficiently route data back to the sink.**

I. Introduction

Clustering in ad hoc and sensor networks develops a suitable platform for efficient data gathering from nodes for reporting back to a sink. This involves grouping a set of nodes together, where a clusterhead (CH) is chosen for the management of other nodes within the cluster, referred to as clustermembers (CM) and the collection of sensor data from its CMs. Furthermore, it is possible for a node to belong to more than one cluster and hence assist in intercluster communication and routing of data to the sink. Such nodes are termed gateway (GW) nodes. Clustering also has the advantages of better addressing than flat routing protocols such as ad hoc on-demand distance vector (AODV) [1] and the dynamic source routing (DSR) [2] as it provides a heirarchical structure to the network.

Some of the well known clustering schemes are the Low-Energy Adaptive Clustering Hierarchy (LEACH) [3] and the Hybrid-Energy-Efficient Distributed (HEED) clustering approach is proposed in [4]. both of which are self-organizing, and distributed protocols. More recent work aiming at an energy-efficient solutions to clustering in WSNs include [5], [6], [7], [8], [9], [10], [11]. In particular, [12] proposes a clustering scheme which is based on data correlation. In this approach, nodes are grouped into clusters based on the similarity of data of the nodes. However the clustering scheme does not take into account energy as a metric in clustering, which is a significant metric in a WSN. In [13], a routing protocol is used which switches transmission power based on the volume of data to be sent. This approach considers flat routing in oppose to hierarchical routing offered by clustering.

It is important to develop a complete protocol which encompasses both efficient clustering of nodes for data gathering and aggregation, and efficiently routing the data to the sink. A complete solution should integrate a sequence of distributed mechanisms for cluster formation and data gathering, that fosters low energy consumption for the limited-energy-resourced sensors, and allows the aggregation of data in a synchronized and distributed manner. This provides a suitable approach to large-scale deployment of sensor nodes with little reliance on centralized control.

In this paper we propose to develop a self-organized cluster formation and routing protocol for wireless sensor networks (WSNs) with variable data sizes for the purpose of efficient data dissemination. In a self-organized system, also a characteristic of biological systems [14], has a simple characteristic where all entities follow the same rules and react and determine their states and future behavior based on information obtained from their local environment. As a result an emergent property appears which meets the total system's objective, whether it is to build a nest, or to simply preserve the survival of the colony. Furthermore the total system's objective does not depend on any single individual member. Hence such a simple concept is adopted into the proposed protocol, where all nodes follow simple common rules purely based on their local environment and determine their roles in the network. As a result the desired function of efficient data dissemination emerges at a global network level, via a distributed manner without centralized control.

The application of the proposed clustering and routing protocol is for quasi-concurrent data reporting of all sensors back to the sink. This means that nodes are required to report their data to the sink periodically at almost the same time. Hence the protocol is suitable for time-driven rather than event-driven sensor applications. Furthermore, we consider sensors having variable data sizes to send to the sink.

Although generally it is presumed that sensor nodes initially have the same size of data to send across the network, it is possible (and more efficient) to consider scenarios where the sensor nodes posses different data sizes for reporting and can benefit from this property to more efficiently disseminate appropriate and necessary data back to the sink. This can be described in the following scenario. A number of sensors are deployed across a field where each sensor monitors and periodically reports the change in value of a number of observed parameters. For instance, a sensor may periodically report the change in temperature, pressure, and humidity from the previously reported value. A sensor node whose observation have remained the same as the previously reported value(s) would then have a smaller data size to report back to the sink. The largest amount of data is when all parameters have changed, and the smallest amount of data is when no changes have occurred in any of the parameters. The effect of node data size would also become more apparent with nested parameters, where a change in one parameter would require reading of several other parameters. The data size may further increase depending on the nature of data and reporting interval. Such a scenario is an example of live reporting. A sensor may also monitor and store several readings of changed values at various intervals and report them in the future. In this case, nodes for which their relative environmental condition is changing more rapidly than other nodes would have more data to report at the instance of reporting.

The basic mechanism of the proposed protocol dubbed Data-Energy-Clustering and Routing (DECRO) is to initially establish the relative distance of nodes to the sink, followed by a self-organized clustering of nodes, self-determination of relative cluster positions and cluster roles in the network and finally data gathering and routing of aggregated data towards the sink in an emergent systematic manner. The described mechanisms occur subsequently in four phases which will be described throughout this paper.

Though data aggregation is assumed in this paper to reduce the data size at the CH, the paper does not directly treat with data aggregation techniques and the interested reader is kindly referred to [15]. A more recent data aggregation technique is presented in [16], which claims to achieve high compression ratios whilst using lower memory and computational costs. In our simulations we simulate data aggregation as percentage reduction of data and investigate its effect on the proposed protocol.

The paper is structured as follows. Section II takes an overview of related work on related clustering schemes. Section III introduces the proposed clustering and routing scheme, and describes in detail the four phases of the protocol. Section IV shows simulations of the proposed protocol followed by conclusions and future work in Section V.

II. Proposed Clustering and Routing Protocol

A. Overview of Protocol and Assumptions

The proposed DECRO protocol is purely distributed and all nodes simply use their one-hop local information obtained from their one-hop neighbors to determine their function in the network. The network assumptions are as follows: Nodes are quasi-stationary, i.e. are generally static. All nodes have the same transmission capability and transmit at the same power and hence, range. All nodes have unique identifiers (IDs). All nodes have the same capabilities in hardware and software alike. Nodes are uniformly distributed. Each node may have different energy levels and different data sizes to send. All nodes are capable of CH status and able to perform data aggregation for data size reduction prior to transmission. Nodes may perform aggregation once they become CHs. All nodes need to report their data in a quasi-concurrent manner back to the sink. No CH is within communication range of any other CH. Intercluster routing is done by mediation of GW nodes. The required data reporting interval to the sink is long enough for the proposed protocol to perform the required four phases, which will be explained in the following sections. Joining clusters is performed in a passive manner. All non-CH nodes will join all clusters within range.

Similar to previous sensor network routing and clustering protocols, the primary aim of this protocol is maximizing the sensor network lifetime by avoiding node energy exhaustion, through careful design of the proposed multi-phased protocol mechanisms. There are four phases in the proposed integrated clustering and routing protocol. These phases ensure that the cluster formation, data gathering and routing are performed in a coherent manner, without conflicting each other. The four phases are: 1) Hopcount-to-sink initialization and neighbor information discovery, 2) Cluster formation and cluster role determination, 3) Data gathering within a cluster, and 4) Routing of aggregated data from CHs to sink.

The first phase initiates the hopcount-to-sink (hops) of nodes and exchanges node energies and data sizes in order to calculate the relative cost of becoming CHs. Phase 1 triggers phase 2, where cluster formation takes place. At the end of phase 2, clusters identify their roles in the network. There are two main roles for clusters: wait for data from higher cluster, or initiator of routing data, which are the role of the clusters furthest away from the sink, termed highest clusters (HCs). Following phase 2, the data gathering phase (phase 3) begins, whereby the CH collects data from its CMs. Finally the routing phase (phase 4) begins by the initiation of routing by the HCs in the network, followed by intermediate clusters towards the sink. During this phase, the relative position of clusters plays a vital role on how routing is performed.

Figure 1 illustrates an example of clusters formed in a WSN using the proposed DECRO scheme. The figure also shows the roles of clusters. The HCs are typically formed at the edge of the network and are in charge of initiating the routing procedure. These clusters have the highest hops CHs in their region. The nature of such clusters is that downstream data from such clusters would traverse intermediate clusters throughout the network subsequently, resulting in an energy-efficient mechanism that allows clusters to aggregate data from higher clusters and forward this to the sink. CHs which have neighboring clusters having CH hops greater than them, termed higher neighboring clusters (HNCs), will wait for data from these HNCs before aggregating and sending data towards the sink. Such clusters having one or more HNCs are called lower neighboring clusters (LNCs). A CH of a LNC realizes the existence of a HNC when one or more of its GWs belong to another cluster with a CH of a higher hops than itself. HNC is purely relative. An HNC of an LNC itself becomes a LNC

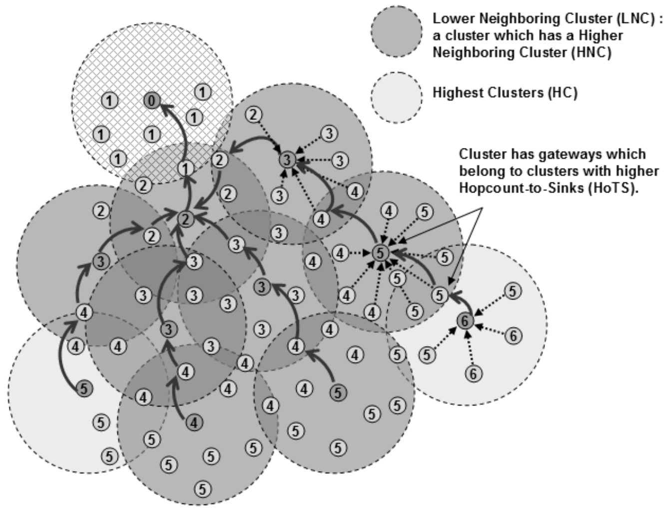

Fig. 1. DECRO cluster formation.

of its own HNC i.e., a CH with a hops of k belongs to a HNC of a cluster hops of $k-1$, however a LNC with CH hops of $k+1$.

In Fig. 1, there are two HCs in the network with hops of 5 and 6 respectively. The initiation of the routing phase by the HC is essential for the purpose of intercluster aggregation. Furthermore, the waiting period of LNCs for data from HNCs mitigates the *funneling effect* [17] that causes clusters and nodes closer to the sink to lose more energy due to the excess forwarding of data from nodes and clusters further away from the sink. A LNC which does not have an $appointed$ GW, i.e. a GW which will forward data from a HNC also initiates routing, as shown in Fig. 1.

We note that the first two phases, occur either once or when needed e.g. due to topological changes and node failure/replacement, whereas phase 3 and 4 are repeated periodically for consecutive data gathering and routing to the sink.

B. Hops Initialization and Neighbor Information Discovery Phase

Initially the sink broadcasts a hops information message (HOPIM) to the entire network. The HOPIM format as $< N_{ID}, hops, energy, dataSize, reporting\_interval >$. The first field is the ID of the node (N) broadcasting the message. The second field is used by the nodes to initialize their relative distance in terms of number of hops from the sink. The energy field specifies the current residual energy of the node, and the dataSize field the current volume of data that needs to be transmitted.

C. Cluster Formation Phase

1) Cluster Formation Algorithm: Clustering is triggered by the first phase. Nodes proceed with the self-organized cluster formation algorithm as follows:

1) After a fixed period t_w for collection of HOPIMs, each node calculates its own cost, as given in (1).

$$Cost_i = e_{max} - e_i + (u_i + u_t) d_{agg}^{-1} E_T + u_t E_R, \quad (1)$$

where e_{max} is the maximum possible node energy, e_i is the residual energy of node *i*, u_i is the current data size of node *i* which needs to be transmitted, u_t is the total amount of data in bits that would be received by node *i* from its neighbors upon becoming a CH, calculated by summing the dataSize fields of all collected HOPIMs from neighbors, d_{agg} is the aggregation factor, E_T and E_R are the energy per bit for transmitting and receiving respectively. It is assumed that all values have been appropriately normalized. We note that if $e_i < (u_i + u_t) d_{agg}^{-1} E_T + u_t E_R$, then the node would not be able to successfully receive and transmit all the necessary data at the current time. The smallest cost node is a good candidate as a CH. Once the cost is calculated, the node sets off backoff timer t_c proportional to its cost, given in (2), and backoff timer $t_{c(max)}$, which expires at a predefined maximum waiting period that determines a node's status, i.e. CH, GW, or CM. The main purpose of the $t_{c(max)}$ timer is for allowing the reception of multiple claims from different CHs for nodes to obtain possible GW status. The two timers are set off simultaneously.

$$t_c(i) = t_{c(max)} \frac{Cost_i}{Cost(max)} + \eta, \quad (2)$$

where $t_c(i)$ is the waiting time of node i before it attempts at electing itself as a CH, $t_{c(max)}$ is the maximum waiting period for node status determination, and $Cost(max)$ is the maximum possible cost of any node, η is a random jitter introduced to minimize collision probability when nodes have the same energy and data sizes, by reducing the probability of the nodes transmitting at the same time when their cost is very close to each other. The maximum cost can be derived in various ways, including statistically, however a simple way is by setting parameters in 1 as follows. u_t is the total sum of data from all nodes in the network, d_{agg} is set to one, and u_i is set to the maximum size of data a node can possess at any time for transmission.

2) Since the backoff timer t_c of the smallest cost node in the vicinity is more likely to expire first, the smallest cost node will elect itself as CH and send a CH claim (CHC) message to its neighbors, preventing the neighbors from obtaining CH status. The CHC message has the format of $< CH_{ID}, hops(CH_{ID}) >$, containing the ID of the CH and its corresponding hops value.
3) If two nodes within range produce a CHC simultaneously, the node with the lower ID will give up its CH status by broadcasting a CH Declaim (CHD) to its one-hop neighbors in order to dissociate neighbors that have associated with it.
4) The neighbor nodes receiving the CHC message will cancel their t_c timer, and allow $t_{c(max)}$ to expire for

collecting other possible CHC messages from neighboring clusters, prior to the initiation of reporting their data (phase 3). When a member receives CHC from more than one CH, it becomes a GW candidate for intercluster routing of data to the sink. Furthermore, from the hops field of the CHC messages, the candidate GW discovers its lowest hops CH which it will unicast its data to at the time of data reporting in phase 3.

D. Data Gathering Phase (Intra-Cluster)

In parallel to the clustering phase, the data gathering (phase 3) is initiated. This phase is triggered upon a node's timer reaching $t_{c(max)}$. CMs, including GW nodes report their data to the CH by unicasting their data to their CH in a Cluster Member Data message (CMD) with the format of $< N_{ID}, CH_{ID}, hops(CH_{i...n}), data >$, where N_{ID} is the ID of the source CM, CH_{ID} is the ID of the target CH the message is being unicasted to, and the $hops(CH)$ field include the hops of all the CHs a GW (CM belonging to multiple CHs) belongs to. This information is used by the overhearing HNC CHs for discovering the GWs to LNCs. Hence although the HNCs ignore the data of the CMD, they extract and store the N_{ID} and hops of CHs in the CMD message. If a node is a GW, it will unicast to its lowest hops CH as shown in Fig. 1 (dotted arrows). If the GW node belongs to CHs of equal hops, it will choose the one with the higher residual energy, else chooses one randomly. This mechanism is predominantly used to avoid identical data being sent to two or more clusters. This not only reduces redundant data, but also other energy-wasting overhead performed by the additional and unnecessary CH(s). The CHs wait for some time for data to be received by the CMs. This waiting time depends on the total amount of data in the cluster, number of nodes in the cluster, and other MAC protocol-related influences, which is beyond the scope of this paper. However for the sake of simplicity, we can assume that initially the waiting time of the CH is the total size of data from all members of a CH which will be sent to the CH, obtained from HOPIM messages, divided by the effective bandwidth. CHs gather the data from their CMs and aggregate this data for reporting to the sink.

E. Routing Phase

Once the data gathering is complete by each CH, intercluster routing of data to the sink can begin. If a CH does not have a HNC, i.e. it is a local HC, it will immediately unicast it's aggregated data message to its highest energy GW which leads to the lowest-hops LNC. It is important to choose the highest energy GW among the lowest hops GWs, i.e. hops precedes energy for selection, as otherwise a higher energy GW of a higher hops takes a longer path, which will consume more network energy, in addition to further data accumulation by additional clusters being traversed as a result of the longer path. If a CH CH_i does have a HNC, in the first round of routing, it will wait for a period $t_{hnc}(CH_i)$ to receive data from higher clusters before further aggregating and routing towards the sink. In such a case, the appointed GW will also wait for t_{hnc} to obtain HNC's data from CH_i. t_{hnc} must be sufficiently long enough to ensure the lower clusters towards the sink will receive the aggregated data from higher clusters in the first round of routing. Hence t_{hnc} should take into account MAC protocol-related delays as well as propagational and processing delays. We assume the time take to cross clusters in a typical large network is lower than the reporting interval, hence an appropriate value for the initial t_{hnc} is to set it to the reporting interval. Once this timer expires, the CH will forward its own aggregated data towards the sink. Furthermore, t_{hnc} is reduced after receiving the first data packet, allowing nodes to sleep during the waiting period of HNC's data. The process continues until the sink receives all the data.

Before routing of data can begin, HNCs need to be determined, and best GW for each cluster needs to be identified.

1) Highest Cluster Determination: When CH_1 hears a GW's CMD message containing a CH hops greater than itself, it knows that a neighboring higher cluster (HNC) exists and so itself cannot be a HC. If CH_1 does not receive a GW CMD message containing a CH hops greater than itself, it assumes it is a local HC, and so in charge of initiating the routing phase. All non-HCs will proceed with further aggregation and rebroadcasting upon receiving the aggregated data from their HNCs and forwarding the aggregated data towards the sink via their LNCs.

2) Best Gateway Determination: For the purpose of routing (phase 4), energy becomes the primary factor. Hence, out of the smallest hops GWs, the GWs that have the highest amount of residual energy after forwarding the data are chosen to route the data back to the sink. A CH calculates the residual energy of its individual GWs after the GWs have transmitted their data to their CH, and uses this prediction to determine the highest energy GW for routing, using the following:

$$E_{ic} = e_i - e_{CMD} - u_i E_T, \tag{3}$$

E_{ic} is the current energy of GW *i*, e_i is the energy of GW i as obtained from the HOPIM exchanged during phase 1, e_{CMD} is the energy used up to transmit the CMD message, and u_i is the size of the data that was transmitted by GW *i* obtained through the HOPIM. If GW i is chosen by the CH to forward its data, e_i stored at the CH is replaced with E_{ic} for the next routing phase. Once the CH discovers the best GW, it will send a GW appointment (GWA) message, in the format of $< CH_{ID}, GW_{ID} >$ denoting the ID of the GW to be appointed by CH_{ID}. This forces the GW to become an Appointed GW (AGW) and hence stay awake in order to receive the aggregated data from the CH and forward it to the LNC. Other candidate GWs simply act as normal CMs and go to sleep upon sending their data to the CH in phase 3. The routing is initiated by the CH of the HC, where the CH unicasts its aggregated data to its best GW using a Node Aggregated Data message (NAD), with the format of $< S_{ID}, D_{ID}, aggregatedData >$, where S_{ID} is the ID of the source node, in this case the CH, and D_{ID} is the ID of the destination node, in this case the GW in which the CH

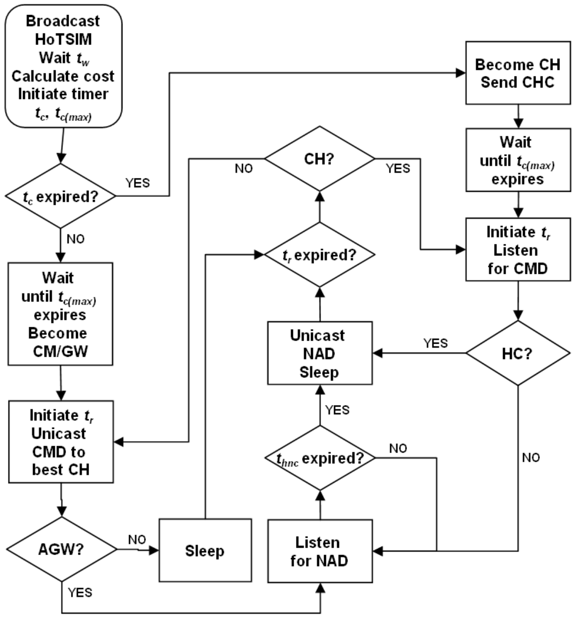

Fig. 2. Flowchart of DECRO mechanisms.

is unicasting its aggregated data to, to be forwarded to the next LNC. The GW is identified by the CH from the CMD messages received by the CH during the data gathering phase. A GW that receives the aggregated data from one of its CHs will unicast it to the appropriate next CH towards the sink in a new NAD message destined for the next CH. The aggregated data is not modified in any way by the GW, and is identical to that obtained from the NAD message of the previous CH. The next CH of the LNC, will continue this process by further aggregating the data and forwarding it accordingly in a new NAD message. This process continues until the sink receives the final NAD initiated from the HC.

F. Integrated Protocol Mechanisms

The basics of the multi-phase DECRO protocol mechanisms are illustrated in the flowchart of Fig. 2. Some of the extended mechanisms and features of the protocol, such as reclustering have been left out for simplicity. Fig. 3 describes the timeline of the four consecutive phases. The protocol contains six timers in total which include four predefined timers, namely t_w, t_r, $t_{c(max)}$, and t_N, and two variable timers t_c and t_{hnc}, which are adjusted after the first round of reporting. In the figure, periods, i.e. non-timers, are expressed in brackets. These include transmission period (t_T), sleep period (t_S), reception period (t_R), and data aggregation period (t_{agg}). Timer t_N is used by CHs and appointed GWs for the routing phase. It is a timer used to check whether the data gathering phase is complete for the current round. During t_N, a node is in the state of idle listening. t_N reinitiates after each segment of data is received, and once expired the node assumes that all the data for the current instance of reporting has been obtained, and hence begins the next task, e.g. aggregation or transmission of data.

There are four primary nodes in the figure which take part in the basic functionality of the DECRO protocol. Initially all nodes are normal nodes without any status. The protocol is initiated when the sink broadcasts a HOPIM. After a time t_w of waiting for the reception of HOPIM from neighbors, nodes calculate their costs and setoff the two backoff timers t_c and $t_{c(max)}$. In the figure, nodes j and l have $t_c(j)$ and $t_c(l)$ which expire first in their respective neighborhoods, hence becoming CHs and broadcasting CHCs to their neighbors, and then going to sleep. Upon receiving the first CHC from node j, nodes i and k cancel their t_c timers. At the time of the expiration of $t_{c(max)}$, node k has also received a CHC from node l, and so obtaining a GW status, and node i becomes a normal CM. Upon expiration of $t_{c(max)}$, all nodes setoff their backoff timer t_r. The two CHs wake up to listen for data from the CMs. The CMs sense data, and unicast the information in a CMD message to their appropriate CH. All GW nodes except the GW nodes which received a GWA message from their CH, will go to sleep shortly after transmitting their CMD messages. The appointed GW will remain awake and wait to receive NAD messages from the HNC, in order to forward this to the LNC.

It is important to note that the first round of routing, the CH and appointed GW of an LNC wait for a period of t_{hnc}, and this waiting time is adjusted after the first NAD message is received from the HNC. Upon the reception of the first NAD message, the CH and appointed GW of an LNC can determine the time it takes for the next NAD message to arrive. Hence, for consecutive rounds of phase 4, the CH will sleep after receiving CMD messages from its neighbors, and awakes just before the expected arrival of the next NAD message from the HNC. Nodes always awaken at the time their t_r expires, and reset this timer upon waking.

G. Reclustering

During the lifetime of the WSN, reclustering is necessary. Reclustering may be triggered via several methods, such as the sudden death of a CH. An imbalance of energy/data may also cause reclustering, however, this may require constant exchange of HOPIM between neighboring nodes, and processing of overheard data messages. Hence, actual reclustering frequency depends on several factors, including battery power, data size, and the reporting interval to the sink. Reclustering can be triggered by an appointed GW when it does not receive or overhear a NAD messages for a predefined number of rounds of reporting (a counter is incremented each time a round is complete, determined by the cycle of waking and sleep). The GW will trigger a local reclustering by broadcasting a HOPIM message with a hops value of -1, which limits the broadcast locally, in oppose to the HOPIM message broadcasted by the sink with a hops value of 0 which is broadcasted to and causes reclustering of the entire network. The sink is able to perform reclustering of the entire network at any time by broadcasting a HOPIM message with a hops value of 0. The sink may generally do this after several rounds of data gathering and reporting.

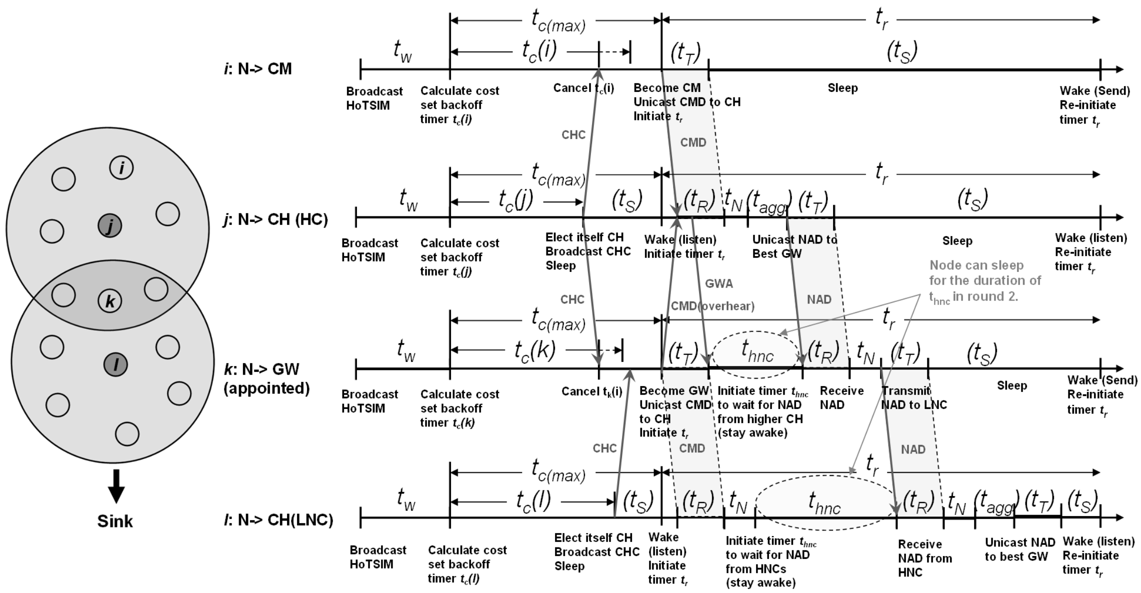

Fig. 3. Complete DECRO timing mechanisms.

III. Simulation

Simulations are performed by implementing the protocol in the Java programming language. To first investigate a typical emerging network that results from the DECRO approach, 500 nodes are uniformly simulated with a range of 90 m across a 500 m by 500 m region. Fig. 4 shows the emerging network that results. The dark clusters represent the HCs, and light clusters are intermediate clusters. In this case a HC is produced at the center of the network as no GWs exist to any adjacent HNCs. This unpredictability is expected in self-organizing systems as nodes are only aware of local information, and have no global knowledge of the network. This is at the expense of robustness and decentralized control. The HCs initiate routing as shown and traverse intermediate LNCs towards the sink. Although most intermediate clusters are traversed via the HNCs as initiated by the HCs, some clusters remain untraversed, in which case their t_{hnc} expires and their data is sent towards the sink independently. In the figure, the arrows show the general direction of routing of data and intermediate clusters being traversed.

Using the sensor node energy consumption model used in [3] we take 500 nJ/bit for the transmitter and receiver circuitry, and 100 pJ/bit/m^2 for the transmitter amplifier. Transmission range starts at 50 m, with 10,000 nodes are arranged uniformly across a rectangular area of 500 m by 500 m. Each node has variable data sizes of up to 8000 bits. Nodes initially have variable energy. In the simulations we assume the dissipated energy is caused by the post-clustering phases (phase 3 and phase 4).

Fig. 5 shows the results for the total energy used up by the network when varying the data aggregation efficiency (defined by the percentage of data reduction for both intra-cluster and inter-cluster aggregation) for different transmission ranges. From the figure, the higher transmission ranges results in higher energy consumption in comparison to that of lower transmission ranges. This is due to the fact that by choosing higher transmission ranges, individual nodes use up more energy to transmit at a higher power. However, data aggregation can significantly reduce the energy consumption of the network as shown in the figure.

Fig. 6 shows the average number of GWs and non-GWs produced per cluster as the transmission range is increased. From the figure, as the transmission range of nodes increases, the number of GWs significantly increases, and although the number of member nodes increases, it does not increase as sharply in comparison to the number of GWs. Although increasing the transmission range results in increased number of GWs, the advantage of this increased number of GW nodes gives CHs more choice for GW selection for both choosing the highest energy GW for routing data towards the sink, and also for alternating between several GWs at each round of data gathering and routing. We also note that the whether a node is a GW or a non-GW member does not deteriorate performance in any way as GW nodes also send their data to only one CH.

Fig. 7 shows the relative energy consumed by all CHs, GWs, and non-GWs, after 30 data gathering and reportings to the sink. All nodes in this scenario have a data size of 4000 bits. From the figure, energy consumed by non-GW members is negligible in comparison to that of the energy used up by the CH and GW nodes. Furthermore the energy of non-GW nodes is not affected by data aggregation, as they simply send their own data and not affected by any data aggregation. We also note that although using no data aggregation causes the CH to use up more energy than GWs, as the data aggregation ratio increases, the GW energy usage increases. There are two reasons for this. GWs first send their own data then forward

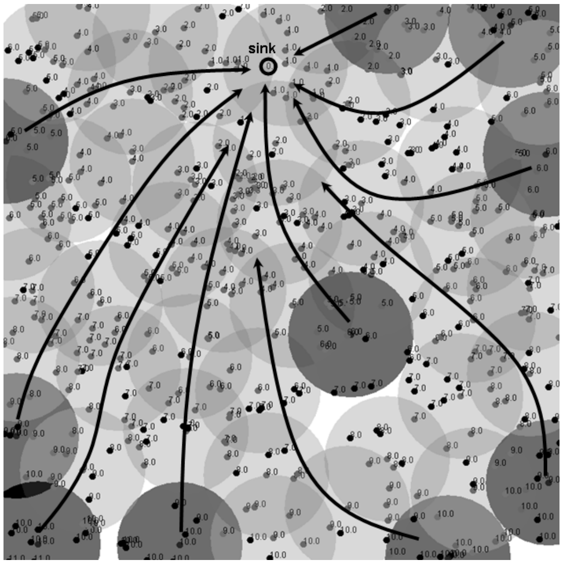

Fig. 4. Emerging network.

the aggregated data of their CH to the next cluster, whereas CHs withhold their own data, aggregate it with the data of their members, then forward it to the GW. Although the total energy of GWs used through several rounds of reporting is higher than CHs, individual GWs use up less energy than their corresponding CHs, as the same clusterhead has to repeatedly repeat data gathering and forwarding, whereas the GWs alternate in forwarding data. This is because the CH generally has several GWs (more as density of nodes and transmission range increase) to choose from for routing its data, and it will send it to the one with the (lowest HoTS) highest energy GW.

DECRO is simulated against a modified version of HEED dubbed (HEED-ER) - HEED - Enhanced Routing. This version of HEED routes data to the sink via the highest hops, highest energy CH, and hence has an enhanced performance over the use of traditional routing protocols such rather than using traditional approaches such as DSR [2] and Directed Diffusion [18] as suggested in the original paper of HEED [4], which are more prone to flooding the network. The reporting is performed periodically. The simulation compares pure HEED-ER and DECRO in regards to the total energy consumed by the network with number of reports to the sink. In the simulation we set the clustering range to 390 units for DECRO and 390 units for HEED clustering range and 555 units for intercluster communication range. Fig. 8 shows the result for when no aggregation is used. In this figure, HEED-ER performs better for the first four reports, however DECRO soon surpasses in performance after the fifth report. Fig. 9 shows the results when 50% aggregation is used. In this figure, DECRO outperforms HEED-ER even from the first report. Fig. 10 shows the result when 90 % aggregation is used. In this figure, DECRO outperforms HEED-ER at even a greater extend compared to the lower aggregation factors. The apparent significant performance of DECRO with

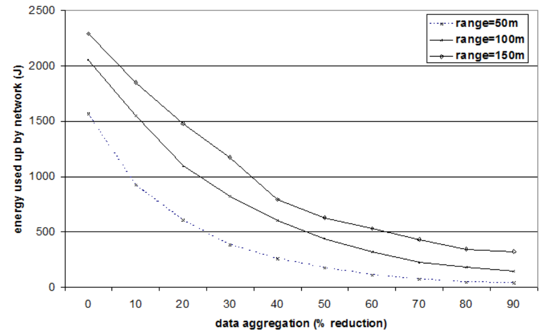

Fig. 5. Total network energy usage with varying data aggregation efficiency.

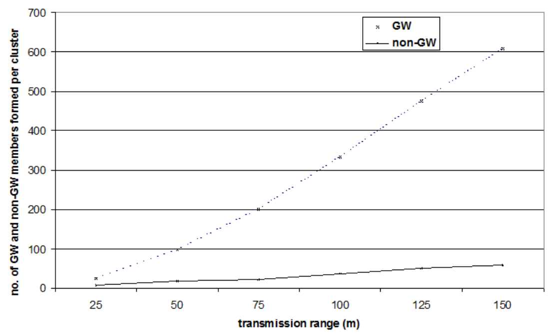

Fig. 6. Average number of gateways (GW) and non-gateway (non-GW) members formed per cluster with varying transmission range.

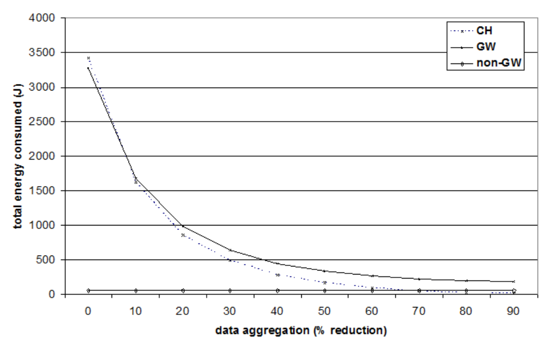

Fig. 7. Relative energy consumption of CHs, GWs, and non-GWs.

increasing aggregation factor is due to the fact that intercluster aggregation is possible in DECRO, whereas this is not possible in HEED-ER.

IV. Conclusion and Future Work

In this paper we introduced a self-organized clustering and routing protocol which works in a distributive manner. Several phases constitute the proposed clustering and routing scheme, which effectively performs data gathering from all sensor nodes within the network in a quasi-concurrent manner and routes the data back to the sink in a multi-hop fashion. The protocol works on the idea of initialization of reporting by the highest clusters in the network, which identify themselves in a distributive and self-organized fashion, and the subsequent

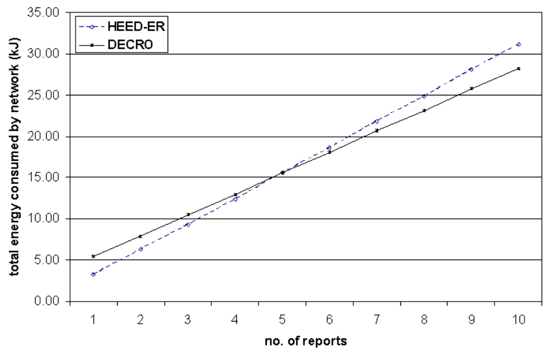

Fig. 8. Total energy consumed vs number of reports for HEED-ER and DECRO (zero aggregation).

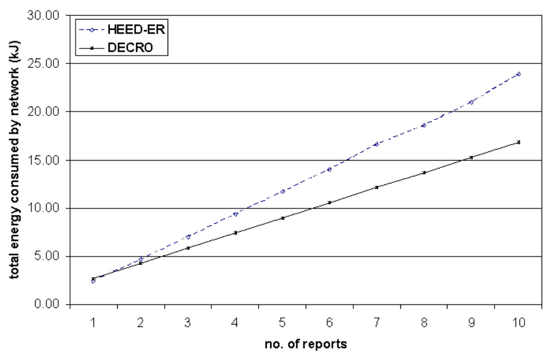

Fig. 9. Total energy consumed vs number of reports for HEED-ER and DECRO (50% aggregation).

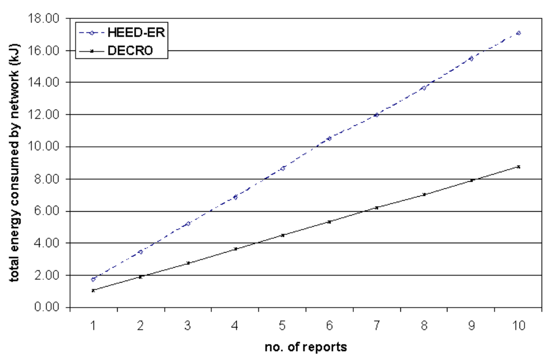

Fig. 10. Total energy consumed vs number of reports for HEED-ER and DECRO (90% aggregation).

aggregation of data by intermediate clusters towards the sink. Simulation results show the effectiveness of the approach in reducing the amount of energy consumed by the network. In this paper MAC-related issues have been left out for simplicity and to present the main concept of the scheme, however although the scheme is independent of the MAC protocol, the actual performance is affected by the underlying MAC protocol used, hence this should be studied in detail in future work.

ACKNOWLEDGMENT

This research was supported in part by the Global COE (Centers of Excellence) Program of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

REFERENCES

[1] C. Perkins, E. Royer, and S. Das, “Ad hoc on-demand distance vector (aodv) routing,” *RFC 3561*, 2003.

[2] D. Johnson, Y. Hu, and D. Maltz, “The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4,” *RFC 4728*, 2007.

[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proc. of the 33rd Annual Hawaii International Conference on System Sciences (HICSS’00)*, Hawaii, USA, Jan. 2000, pp. 3005–3014.

[4] O. Younis and S. Fahmy, “Heed: a hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 3(4), pp. 366–379, Oct-Dec. 2004.

[5] M. H. Tolou and J. Chitizadeh, “Lifetime prolonging of wireless sensor networks via a recursive clustering algorithm,” in *Proc. of the third IEEE International Conference in Central Asia on internet the Next generation of mobile,wireless and optical communications networks (IEEE/IFIP ICI’07)*, Tashkent, Sep. 2007, pp. 1–6.

[6] C. H. and M. S., “Cluster sizing and head selection for efficient data aggregation and routing in sensor networks,” in *Proc. of IEEE Wireless Communications and Networking Conference (IEEE WCNC’06)*, vol. 4, Las Vegas, NV, USA, 2006, pp. 2318–2323.

[7] M. Ye, C. Li, G. Chen, and J. Wu, “Eecs: An energy efficient clustering scheme in wireless sensor networks,” in *Proc. of the 24th IEEE International Performance, Computing, and Communications Conference (IPCCC’05)*, Phoenix, Arizona, USA, Apr. 2005, pp. 535–540.

[8] Y. Zhou, M. Hart, S. Vadgama, and A. Rouz, “A hierarchical clustering method in wireless ad hoc sensor networks,” in *Proc. of the IEEE International Conference on Communications (IEEE ICC’07)*, Glasgow, Scotland, Jun. 2007, pp. 3503–3509.

[9] R. Krishnan and D. Starobinski, “Efficient clustering algorithms for self-organizing wireless sensor networks,” *Ad Hoc Networks*, vol. 4, no. 1, pp. 36–59, Jan. 2006.

[10] J. Kamimura, N. Wakamiya, and M. Murata, “Distributed clustering method for energy-efficient data gathering in sensor networks,” *International Journal on Wireless and Mobile Computing*, vol. 1, no. 2, pp. 113–120, 2006.

[11] J. Yu, W. Liu, J. Song, and B. Cao, “Eemr: An energy-efficient multi-hop routing protocol for wireless sensor networks,” in *Proc. of the International Conference on Computer Systems and Application (IEEE/ACS AICCSA’08)*, Doha, Qatar, Mar. 2008, pp. 291–298.

[12] M. H. Yeo, M. S. Lee, S. J. Lee, and J. S. Yoo, “Data correlation-based clustering in sensor networks,” in *Proc. of the International Symposium on Computer Science and its Applications (CSA’08)*, Hobart, Australia, Oct. 2008, pp. 332–337.

[13] R. Gunasekaran and Q. Hairong, “Xlrp: Cross layer routing protocol for wireless sensor networks,” in *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC’08)*, Las Vegas, USA, Arp. 2008, pp. 2135–2140.

[14] S. Kauffman, “At home in the universe: The search for the laws of self-organization and complexity,” *AIP Press*, 1995.

[15] R. Rajagopalan and P. Varshney, “Data-aggregation techniques in sensor networks: a survey,” *IEEE Communications Surveys and Tutorials*, vol. 8, no. 48–63, 4th Quarter 2006.

[16] F. Marcelloni and M. Vecchio, “A simple algorithm for data compression in wireless sensor networks,” *IEEE Communication Letters*, vol. 12, no. 6, pp. 411–413, Jun. 2008.

[17] J. Li and P. Mohapatra, “Analytical modeling and mitigation techniques for the energy hole problem in sensor networks,” *Pervasive and Mobile Computing*, vol. 3, no. 3, pp. 233–254, Jun. 2007.

[18] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” in *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM’00)*, Boston, USA, Aug. 2000, pp. 56–67.