Cross-Layer Data-Gathering Protocol for Wireless Sensor Networks

Mohammed A. Abuhelaleh[1], Khaled Almi'ani[1,2], Bassam A. Alqaralleh[1], Moha'med Al-jaafereh[1], and Tahseen A. Al-Ramadin[1]

[1,2] College of Information Technology, Al-Hussein Bin Talal University P.O. Box (20), Ma'an - Jordan
[2]Corresponding author: E-mail: k.almiani@gmail.com

ABSTRACT

Wireless Sensor Networks (WSNs) are normally deployed in hostile environments (hard-to-access). This introduce reducing the energy consumption as a major challenge, since recharging the sensors batteries is impractical. In this paper, we further reduce the energy consumption, we present an extended version of the RMC [1] protocol. The goal of RMC is to contribute towards avoiding the overhead caused by obtaining the transmission schedule. RMC introduce a dependency relationship between the behavior of the forwarding and each node. Thus, each node can locally determine its schedule and forwarding path. We evaluate the performance of our approach through simulations and the results show that our approach achieves significant performance gains over other approaches.

Keywords: Wireless sensor networks, Cross-Layer Data-Gathering Protocol

1. INTRODUCTION

Sensor nodes are usually battery- powered and the environment in which these nodes are deployed may be hostile. In hostile environments, recharging the nodes is generally impossible. Even in a friendly environment, the sheer size of a network can make recharging impractical. Therefore, the most important metric affecting network performance is the node's residual energy. A node may run out of energy prematurely, thereby leaving an unobserved area.

Current research has focused on energy consumption as a primary concern in WSNs. Numerous factors have been identified as potential causes of energy wastage. These include, idle- listening, collision, control packets overhead, and unnecessarily high transmission range. Many approaches have been proposed to conserve energy. Some approaches target the performance of a single layer whilst others adopt a cross-layer approach in an attempt to achieve better performance.

Considerable attention has been devoted to the data-link and routing protocols to address the above issue. Towards this end, multi-hop transmission and clustering are important design criteria. As the cost of transmitting one bit is higher than the processing cost [3], organizing the network into clusters is vital from an energy conservation perspective. Cluster heads (CHs) collect data from their cluster to estimate data correlations mainly related to spatial proximity and subsequently aim to remove redundant transmission and lower cost.

Various clustering algorithms have been proposed in the literature, and a survey of these algorithms can be found in [5]. The role of these algorithms is to select CHs in order to distribute the load and increase the network lifetime. Some algorithms use probability-based strategies to elect the CHs, where a node is selected as CH based on certain probability distribution. One possible drawback of this election strategy in a multi-hop environment is, the resultant network instability due to the regular

election process that changes the topology. This forces the routing protocol to continuously adapt to such topological changes.

Normally, WSN applications is either event-driven, periodic-monitoring and object tracking [5]. In an event-driven application, a node will not send data until it senses a specific event. In periodic-monitoring, transmission occurs based on interval of time determined by the user In object tracking, the network monitors the movement of an object within the monitored area.

In periodic-monitoring applications, the number of packets that each node sends is an order of magnitude higher than in any other application domain. Thus, clustering and in-network processing for such applications has the best possibility of increasing the lifetime of the network. However, in multi-hop communication paradigm, the use of traditional routing protocols that aim to establish a routing path to the sink may be impractical. Moreover, under some circumstances where clustering is also used, the clustering setup triggered periodically further compounds the cost of re-establishing such routes. Eliminating this routing protocol overhead may be achieved by using geographic routing protocols which are known to be scalable.

In periodic-monitoring applications, the probability of having collision in the network is mainly affected by the traffic and the topology of the network. To reduce the possibility of collision, integrating the MAC and routing layers has emerged as promising approach. This integration involves the performing of the scheduling during the establishment of the routing paths. Several proposals have adopted this approach. For instance, the scheduling algorithm in [5] allows each node to include the available scheduling time interval inside the routing request message. Such an approach seems to be appropriate for sensor nodes as they are generally considered to be stationary.

Another scheduling technique is to use a centralized node to schedule the transmission. For instance, in [7], the authors proposed dividing the nodes based on their distance to the base station(sink). Nodes have the same number of hops to the base station is assigned to the same group. Even though scheduling network transmissions by a single node is a performance bottleneck, its benefits may be realised if a smaller scope scheduling is adopted. This is where node clustering could be useful. Clustering and in-network processing have important design implications on transmission scheduling. The use of clustering affects the scheduling process in two ways. Firstly, it reduces the scheduling scope; CHs handle the intra- cluster communication. Secondly, dynamic clustering however alters the network topology and therefore increases the overhead of managing the inter-cluster communication.

In this paper, we present an extended version of the cross-layer protocol RMC [1]. This extension include additional scheduling steps to further extend the lifetime of the network.

The rest of the paper is organised as follows. In Section 2, we discuss the related work. In Section 3, the details of the extension is presented. In Section 4, the performance of the presented protocol is evaluated. We conclude the paper in Section 5.

2. RELATED WORK

There has been significant research aimed at increasing the lifetime of the network by integrating the network functionalities. In [9], a routing architecture known as Sensor-Supernode-Sink (3S) is presented. The network consists predominantly of sensors with power and memory limitations. The remainder of the network consist of supernodes. Supernodes are more powerful devices than sensors and deliver sensor data to the sink. The routing paths are established between supernodes using a minimum spanning tree. Each supernode's zone is divided into sub-zones, where a master node is elected to gather sub-zone data. Each master node schedules the transmission of nodes that must respond to a specific query. A staggered active/sleep schedule is used to avoid the data-forwarding interruption problem. In each round, the node receives a packet from its master to determine whether it is in the active mode or not. However, performance is impaired as the percentage of nodes participating in the processing task increases.

A random scheduling approach to increase network efficiency is proposed in [10]. The model takes energy usage into consideration. By using Extended Power Aware Random Scheduling (EPARS), this model tries to resolve the collision problem. EPARS uses node ID to generate a sequence of intervals. A node state in each interval alternates between sleep, idle, send and receive. By knowing the neighbor IDs, each node can know the sequence of its neighbors' intervals. Each node performs passive sensing to resolve synchronization conflicts. The hidden terminal problem is resolved by letting the destination node broadcast the send schedule to its neighbors. To avoid problems such as latency and congestion, the routing layer mechanism is performed based on the EPARS' schedules. However, its performance is highly dependent on the topology stability, as frequent changes to the topology causes significant overhead in managing the schedules.

By taking into account the response order for the application query, the authors in [11] attempt to improve the MAC scheduling. Based on this order, the schedule of the transmission are established to avoid the possibility of collision. The proposed approach uses heuristic method to eliminate sub-optimal solutions. In addition, the proposed approach uses randomized algorithm to search for least cost solutions. Deploying multiple queries simultaneously may reduces the efficiency of this approach. The techniques behind the above proposals may be classified into two categories. The first category aims to provide the routing protocol with the MAC knowledge to reduce interference and avid congestion. The second category aims to enhance the MAC protocol performance by considering the routing protocol knowledge. In a typical WSN, the deployed nodes typically contain redundancy to ensure connectivity. As a result, many nodes' data may be highly correlated. Exploiting such correlations underlines the importance of network clustering to reduce the number of transmitted packets. However, by altering the network topology, the adoption of a dynamic clustering scheme penalises the performance by increasing the overhead in controlling the transmission schedules and re-establishing the routing paths. In this work, each node determines its transmission schedule based on its location. Network lifetime is considered here as the primary objective. We further simplify the routing mechanism to adopt a number of forwarding

paths. Without explicit path discovery, different paths are used to enhance the load distribution.

3. THE RMC CROSS-LAYER SCHEME

RMC is designed to prolong the network lifetime, and is able to offer good scalability as well as to provide a time-bound on data gathering. By addressing the periodic-monitoring application characteristics, RMC aims to encompass the routing, MAC and clustering functionalities. To achieve its design goal, RMC consists of two main components, namely clustering and scheduling.

Clustering: takes place periodically. Node clustering is performed in a semi-static fashion across two levels, namely horizontal and diagonal. It is termed semi-static because the CH order is rotated based the ascending order of their node id. It is neither fully dynamic where CH selection is probabilistic nor static where the CHs are fixed at deployment time. The adoption of a semi-static scheme is to simplify the routing process, whereas the two-level clustering endeavors to distribute the load.

Scheduling: RMC scheduling is cluster-based. Based on the ID of its cluster, each node determines its schedule.
We assume the traffic is periodic, and the network model adopts the following assumptions:

- Nodes are stationary, location-aware and time synchronized;
- The network geometry is rectangular;
- The nodes are uniformly deployed;
- We currently considers the centre location for the sink. However, the sink node may be located at any corner or at the centre.

As both horizontal and diagonal levels are adopted, the scheduling scheme uses four phases to alternate flows. The adoption of the two levels clustering combined with the four phases aims to vary the routes of the packets, so the traffic load can be evenly distributed. To understand the principles of the RMC mechanism, we use the illustration shown in Figure 1. Each node will be a member of two clusters, horizontal and diagonal. The horizontal cluster ID is used to address the nodes in phase one and three. In the other phases, nodes will be addressed using their diagonal cluster ID. The routing mechanism employed in RMC works in phases one and three to gather the network packets into the middle row and column respectively, towards reaching the sink. In the other phases, the gathering will be at the diagonal before the sink.

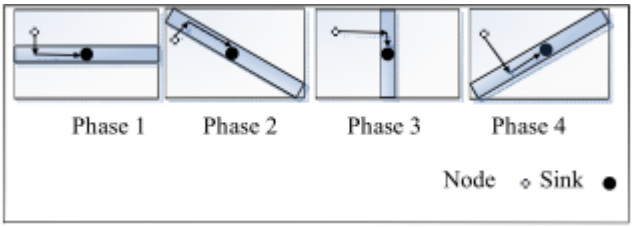

Figure 1: the four phases of data collection with different flow directions. Phase 1 involves data collected vertically through to the middle row and towards the sink. The others follow the direction indicated.

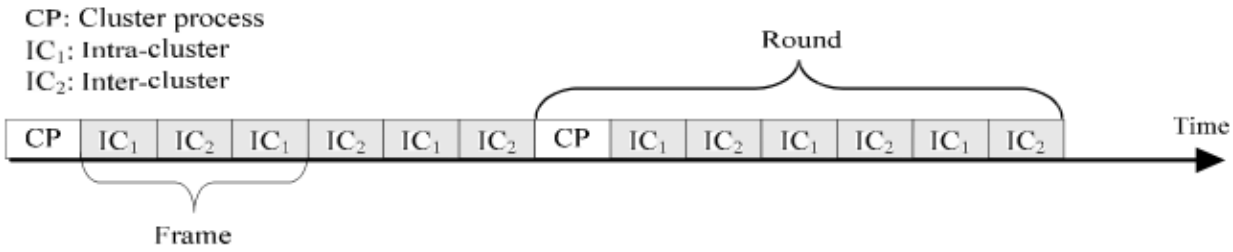

Figure 2: RMC frame

3.1 Clustering Process

RMC adopts a two-step clustering process involving cluster setup and CH election. Its operation is frame-based with a fixed number of slots per frame. Figure 2 illustrates this frame, and round which consists of clustering, intra-cluster communication and inter-cluster communication.

3.1.1 Cluster setup

Several proposals have been introduced to address the routing protocol overhead [7][12]. Generally, they attempt to simplify the routing process by organizing the network into a multi-level architecture. Even though these approaches have achieved significant overhead reduction, their designs have not considered the importance of load balancing. Therefore in this section, we describe the RMC protocol clustering scheme, which is designed to simplify the routing process as well as to promote load balancing. As indicated before, the clustering is performed across two levels, horizontal and diagonal. At each level the network is divided into a number of grids, and each grid host a cluster. For the horizontal level, the clustering process will be performed to obtain m by m grids. The value of m is odd as depicted in Figure 3(a). The odd division ensures that the behavior of the network is symmetric. The size of each grid is determined based on the geometry of the network. On other hand, diagonal clustering divides the network into diamond grids, where each diamond vertex will bisect a horizontal grid edge [Figure 3 (b)]. Using the available information about the network and the grids, each sensor can locally determines its cluster membership. These information can be provided to the network before deployment. To start the process of identifying the cluster, each node maps its location to the relative (x,y) coordinates, where the origin of the network is fixed to the network lower left corner.

To understand how a node locally determines its cluster id, we use Figure 3(c) for exposition. The row's number (r) and the column's (c) number are used to identify a cluster. Accordingly, the horizontal cluster ID is calculated as:

$$r = \left\lceil \frac{x}{h_{cw}} \right\rceil , \qquad c = \left\lceil \frac{y}{h_{ch}} \right\rceil \quad (1)$$

Where h_{cw} and h_{ch} are the horizontal grid width and height, respectively.

In diagonal clustering, a row represents the number of clusters crossed by a line drawn parallel to the diagonal line BD [Figure 3(c)]. Indexing progresses from the lower left corner to the upper right corner. For ease of exposition, we use node H as a reference and draw a line (i.e. EF) through it parallel to BD. To calculate the diagonal row index, we need to determine the intersection point of line EF at the network lower edge. Line EF intersects the network lower edge at F. The row index can accordingly be determined by determining the number of h_{cw} intervals that fits in AF. Similarly, the diagonal column index represents the number of h_{cw} intervals that fits 2XF.
This completes the unique identification of each cluster. The CH election process is given below.

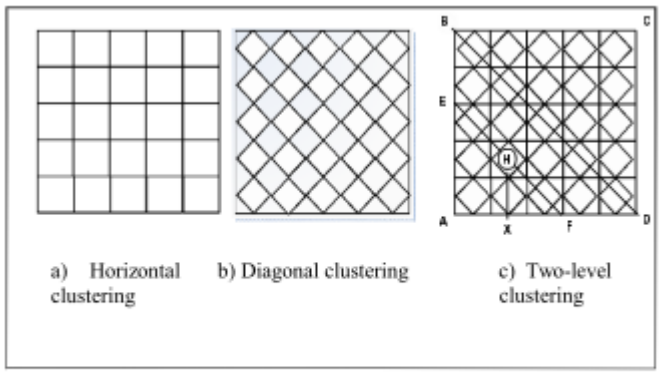

Figure 3: Clustering Process

3.1.2 CH election

The above clustering organises the network into a pre-knownnumber of clusters with the same density. Since the nodes are stationary and each cluster has the same number of nodes, the employed CH election process ensures the load is distributed by rotating the elected CH in sequential order of node id. We refer to a generic 'cluster' on the understanding that this process is used for both horizontal and diagonal clusters. In order to discover neighbours, a node uses a neighbour discovery scheme.Neighbour discovery involves informing each node of itsneighbours. Each node chooses a random slot to broadcast its identity. A slot is the duration needed to transmit one packetdenoted by TX. Thus, the probability that a node and its neighbours choose different slots is given by [13]:

$$p = \prod_{j=1}^{n_c} \left(1 - \frac{j}{s_n}\right) \approx \prod_{j=1}^{n_c} e^{-j/s} \approx e^{-n_c^2/2s_n}$$

wheren_c refers to the number of neighbours of a node and s n is the total number of slots. From the above equation, we can see that the necessary number of slots for p = 0.99 can be calculated as:

$$s_n \approx \frac{-n_c^2}{2 \cdot \ln 0.99} \qquad (3)$$

If n_c is unknown, it can be approximated using the cluster area (A_{cl}) and node density. For intra-cluster communications, the transmission range is set to make sure that the cluster area is coverd. The inter-cluster communications range must ensure

that the next CH can be reached. Consider two adjacent clusters C_1 and C_2 . In the worst case, a CH is situated at the upper right corner of C_1 but its next hop is located at the bottom left corner of C_2 . This will demand high transmission costs. To reduce the probability of such an occurrence, we introduce a procedure termed virtual-clustering.

Virtual-clustering is used to split each cluster into four sub-clusters scl_1 …. scl_4 . Starting sequentially from scl_1 to scl_4 , the node with the smallest ID will advertise itself as the CH.Uponcompletion of a round, the process repeats.

3.2 Transmission scheduling

RMC scheduling is cluster-based. Nodes determine their schedule based on the id of their clusters. As stated above, the process of the scheduling consists of four phases, and in each phase, the scheduling process consists of two stages of gathering; the intermediate gathering and the sink gathering. In the intermediate gathering, the process works by gathering the packets at the middle row or the middle column. Once this step is performed, the process starts the sink gathering step to forward the packets to the sink.

In the intermediate stage the network is partitioned to three partitions P_A, P_M and P_B.. P_M refers to the middle row or column where the packets need to arrive before reaching the sink. In addition, when P_M refers to the middle row, P_A and P_B are the network partitions, where row > P_M and row < P_M, respectively. Otherwise, they represent the partitions where column > P_M and column < P_M, respectively.

In the intermediate stage of phase one, the odd and even column works in alternative fashion to gather the data of the nodes at the middle horizontal row. Phases 2, 3, and 4 largely follow this approach for the scheduling of transmissions. To schedule the packets in phase one without collision, each cluster head must know the number of clusters in its column and the partition to which it belongs. The scheduling logic is determined by the membership of each cluster head partition. For P_A, the following steps are required before scheduling:

1) In Each CH, the number of cluster heads in P_A that can transmit their data collision-free is calculated. This number (C_{ft}) is calculated by dividing the number of clusters in P_A over the minimum number of hops required to separate the non-interfering transmissions. C_{ft} can be calculated as follows

$$C_{ft} = \left\lceil \frac{\left\lfloor n_h / h_{ch} \right\rfloor}{3} \right\rceil \quad (4)$$

wheren_h refers to the network height.

2) Then the number of hops between any cluster and the last cluster in its column is determined (HC).

$$H_C = \frac{n_h}{h_{ch}} - r \quad (5)$$

3) The number of transmissions to perform in one gathering round id determined (T_A). T_A is given by:

$$T_A = \left\lceil \frac{H_C + 1}{3} \right\rceil \quad (6)$$

After determining the number of transmissions, a CH will transmit in one round. Each cluster head transmission can span multiple slots to support a number of packets, thus we represent its period as a transmission duration or TD. Each CH transmission will be assigned to its appropriate time duration. Each time duration represents an interval of time in a cluster's duty-cycle. Each time duration will be occupied by one or more non-interfering CHs. Based on the lowest CH index (i) in TD_1's sequence, the other slot sequences will be determined as follows:

- The first case occurs when i=3.In this situation,CH_1 will be assigned to transmission duration number two. In general the sequence of the transmission duration number i will be awarded as $CH_i = \{CH\ CH_{r-1}\ inTD_{i-1}\}$.
- The second case occurs when the value of i is 2. In this case, the sequence of TD2 will be assigned as $CH_i = \{CH_{r-1}\ CH_r\ inTD_1\}$. Other *TD* will be assigned as $CH_i = \{CH\ CH_{r-1}\ inTD_{i-1}\}$.

In the P_B partition ,the number of transmissions (T_B)that each cluster head performs is calculated based on two values: the number of clusters in the cluster head partition column (PCN), as well as, the cluster head row index (r).T_B is computed as follows:

- If the values of the PCN +1 and r+1 are factor of 3, then T_B value is calculated as $\lceil r3 \rceil + 1$; otherwise the value will be $\lceil r3 \rceil$.
- If r is equal to P_{CN} or its value is a factor of 3, then T_B value will be calculated as $\lfloor r3 \rfloor + 1$; otherwise the value will be $\lceil r3 \rceil$

Figure 4 presents an example to clarify the presented stages. In this example, the intermediate scheduling is for a column with nine clusters.

From the discussion, we can see that the last data to arrive in P_M includes the CH_1 packet. By identifying the *TD*, where the last hop in P_B transmits the flow, we can calculate the amount of time needed to collect the column packets. We expect that the CH_1 data to appears in TD_2 and all subsequent TD_s ,therefore, we can know that the last hop in P_B will always schedule the last flow in TD_{z+1}, where z is the number of clusters in each side. By knowing z, the data gathering time needed of a column towards the P_M cluster is obtained as follows:

$$C_T = (z \cdot (3 \cdot TX)) + 2 \cdot cluster\_time$$

Where *cluster_time*is the intra-cluster time for data collection.

Once the packet are collected at P_M, the process proceed by triggering the sink

scheduling stage to forward the collected packets to the sink. In P_M, each CH's load will be (2z+1) packets (equivalent to the number of clusters in a column).Therefore, eq.(7) can be re- written to represent the time needed to gather the network load to the sink as:

$$N_T = 2 \cdot C_T + ((6z^2 + 3z) \cdot TX) \quad (8)$$

While considering the direction of the flow direction, the scheduling in phase three will be identical to phase one. However, in diagonal scheduling, each CH must undertake several steps before adopting the process described for horizontal scheduling. For instance, in phase two, the steps for a CH will be:

- Determine its partition membership.
- Compute the number of clusters in its row. Using the id of the cluster and PM, a cluster head can find the number of clusters in its row by:

$$cl(r) = (P_M \cdot 2 + 1) - (r - P \cdot 2)$$

The smaller clusters at the beginning and end of each row and column will be handled as regular clusters

.

And the total time needed to gather al lnetwork data at the sink is given by:

$$N_T = 2 \cdot L_C + \sum TD_i \cdot TX$$

Where $\sum$ TD Refers to the duration of all TDs in the sink scheduling and LC refers to the gathering time needed for the longest diagonal column in the network. By changing the flow direction, the same steps can be applied to the scheduling process of phase four. In the following, we demonstrate how the cluster size can be determined to satisfy a given application gathering time bound.

4. SIMULATION FRAMEWORK

The validate the performance of the presented protocol, we used the the J-Sim simulator [14]. Regarding the propagation model, we used the free-space model. We assume that all nodes are required to transmit their data once in an interval of time determined by the end-user. Also, regarding the behavior of the RMC protocol, nodes determine the id of their next hop based on the current active phase.

To benchmark the proposed protocol, we compared it performance against the HEED protocol [17]. This protocol can not be directly used in a multi-hop communication paradigm. Therefore, for the fairness of the comparison, we facilitated the HEED protocol to use both, the GPRS [18] protocol and the AODV [19] protocol, to act as the routing protocols. In both cases, only the CHs are used for forwarding purposes. By investigating against two different routing protocols, we aim to underline the effect of decoupling routing and clustering on the network performance. Moreover, at the data link layer for HEED simulation, the MAC functionality is handled by 802.11 MAC protocol. For the rest of the paper, the acronyms HEED-Loc and HEED-Hop will be used to refer to the routing protocol in use, where Loc refers to the GPSR protocol and Hop to the AODV

protocol. However, to ensure the fairness of the comparison, the effect of the collision in intra-cluster communications is eliminated.

5. RESULTS AND DISCUSSIONS

Our underlying goal is to investigate the performance in terms of the energy consumption and the lifetime of the network. We define the lifetime of the network, as the time until the first sensor node run out of energy. Also, we assume that the nodes are uniformly distributed across N × N m2 where N. = 300 m. The width and height of the horizontal clusters are 100 m. In HEED, the transmission ranges for the intra-cluster and inter-cluster are 60 m and 150 m. To balance the load, node degree is selected as the secondary parameter. The cluster round is selected to be 100 sec. In AODV and GPSR, the lifetime of any path is set to 100 ms. All the following experiments adopt these transceiver energy parameters: Eelec = 50 nJ/bit and Eefs = 10 pJ /bit/m^2. We vary the sensing interval, number of nodes and radio range. The duration of each simulation experiment is 50 minutes. The shown results is the average of 50 runs and we draw 95% confidence intervals. From the earlier description, it might be obvious that RMC components are designed without any stochastic behavior. As a result, in the following experiments, its performance will be plotted without confidence interval since repeating the experiment has no effect on its behavior.

In Figure 5, for a network with 100 nodes, we show the impact of the sensing interval on the average energy consumption . It is evident that RMC is able to achieve substantial reductions in energy consumption compared to HEED-Hop and HEED-Loc. The major factor behind the difference between their behavior is the routing mechanism. HEED-Hop establishes the routing path among the CHs like a tree. However, in each cluster round upon CH re-election, the routing path needs to be re-established causing a significant overhead. Adopting such routing techniques in periodic-monitoring applications will most likely lead to congestion; this is due to high probability of having more merged paths at lower levels of the tree. On the contrary, HEED-Loc overcomes this problem by establishing routing paths depending on an individual node location. However, in each cluster round, HEED-Loc requires re-performing the new neighboring CHs discovery. RMC simplifies the routing process by establishing the CH communications based on cluster IDs. The identity and the proximity of a cluster remain the same in all cluster rounds. Therefore, due to the elimination of the routing protocol overhead, RMC is the most energy efficient protocol here.

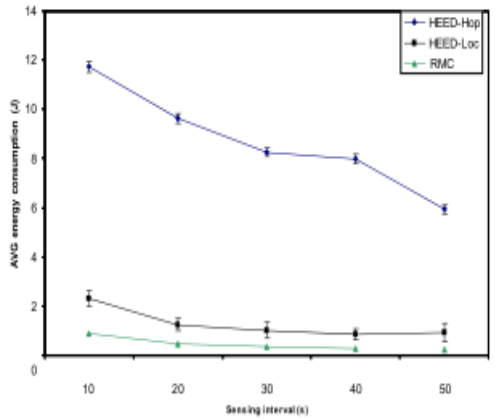

Figure 5: Average energy consumption against sensing interval

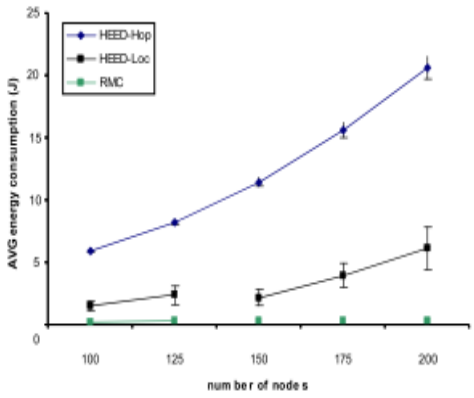

Figure 6: Average energy consumption against number of nodes

In contrast, Figure 6 shows the relationship between number of nodes and average energy consumption (sensing interval 50s). The results show that the RMC protocol constantly outperforms the other protocols. The RMC protocol achieves a near constant energy usage compared to the exponential rise in energy dissipation for HEED-Hop and HEED-Loc. The key contributing factor in this situation is the adopted clustering mechanism. For HEED, the number of generated clusters is largely dependent on the number of nodes. Increasing the number of nodes raises the collision probability during the cluster setup phase. As a result, the number of generated clusters increases. This affects the performance by increasing the network traffic and the overhead of managing the routing protocol functionalities. In RMC, increasing the number of nodes only has minimal effect on its performance.

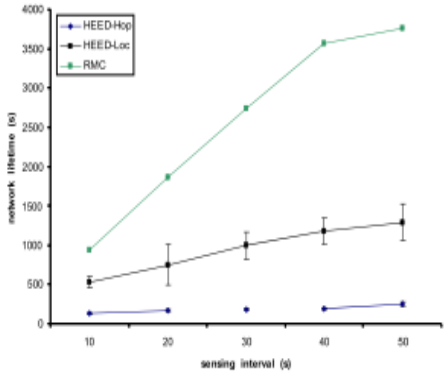

Figure 7: Network lifetime against sensing interval

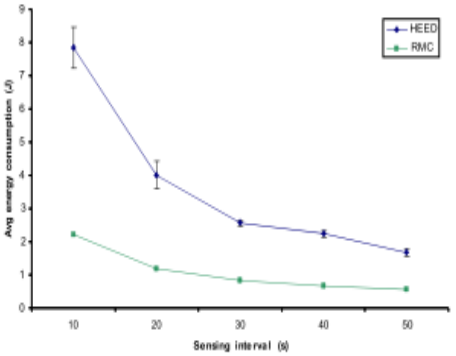

Figure 9: Average energy consumption against sensing interval for a single broadcast domain network.

To investigate how the sensing interval influences the lifetime of the network, we ran the simulation assuming each node is initially equipped with 2J. Figure 7 shows the results of this evaluation. For HEED-Hop and HEED-Loc, by increasing the network load it increases the collision probability. However, in HEED-Hop varying network traffic has a minimal impact on its network lifetime compared to HEED-Loc its lifetime improved considerably with reducing traffic. This behavior is mainly due to the high probability of congestion in HEED-Hop compared to HEED-Loc. In RMC, the static clustering in conjunction with the scheduling strategy works to significantly balance the load distribution.

Therefore, besides eliminating collisions and simplifying routing, RMC substantially prolongs the network lifetime.

Figure 8 shows the impact of the number on the lifetime of the network. The sensing interval is set to 50s. from the results, we can see that increasing the number of nodes reduces the difference between HEED-Hop and HEED-Loc. As one would expect, this behavior is mainly due to the HEED mechanism. Increasing the number of nodes leads to increase in the number of generated clusters and therefore increases interference. RMC only experiences liner drop in its network lifetime against the exponential drop seen by the other schemes. This linear drop is due to the fact that by increasing the number of nodes, each node expends more energy due to overhearing, thereby reducing its performance.

All preceding experiments consider RMC's performance in a multi-hop environment, whereas the following experiments evaluate RMC's performance against HEED in a single-hop environment. This study aims to discover efficiency across a single broadcast domain with the routing impact discounted. Figure 9 shows the average energy consumption across the network while varying the sensing interval. It is evident, that RMC outperforms HEED more significantly in heavy traffic compared to light traffic. Two facts contribute to this behavior. Firstly, the overhead involved in cluster setup for RMC is significantly lower than HEED. Secondly, HEED's probabilistic CH election and tentative CH declarations are likely to produce more clusters than RMC. In light traffic, the major factor influencing HEED's performance is the cluster setup overhead, whereas in heavy traffic, the larger number of generated clusters increases its energy consumption.

Finally, to investigate the impact of the sensing interval on the lifetime of the network, we ran the experiment, which fixing the sensors initial energy level to 2J. Figure 10 presents the results of this experiment, RMC is able to achieve an improvement of up to seven times compared to HEED. As expected, this behavior is mainly contributed RMC's two-level-clustering scheme, which achieves good load balancing. Moreover, HEED typically has more number of clusters compared to RMC and therefore contributes to higher amount of network traffic.

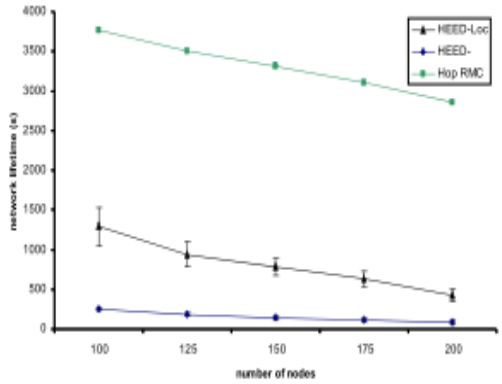

Figure 8: Network lifetime against number of nodes

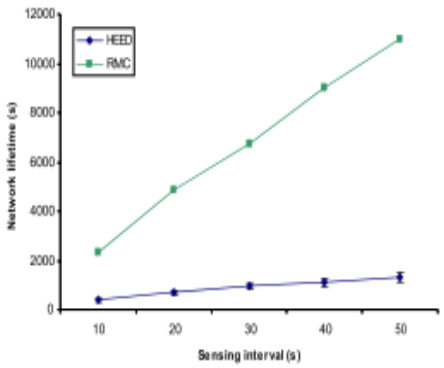

Figure 10: Network lifetime against sensing interval for a single broadcast domain network.

6. CONCLUSION

In this work, we presented an extended version of the cross-layer data gathering protocol RMC[1]. By simplifying the scheduling process as well as the routing mechanism, this protocol aims to eliminate the overhead, which is the result of considering the clustering, MAC, and routing protocol as separate components. Typically, the performance of the scheduling scheme is influenced by the topology structure. This occurs because changing the topology result in re-calculating the schedule. The RMC protocol eliminates this overhead by introducing a dependency relationship between the transmission schedule for each node and its location. RMC adopts two-level clustering with geographic-based routing in order to enhance the load distribution. It also provides the application with a mechanism to control the maximum network gathering time bound. The results show that presented protocol significantly improve the lifetime of the network.

REFERENCES

[1] K. Almiani, S. Selvakennedy and A. Viglas ‘RMC: An energy-aware cross-layer data-gathering protocol for wireless sensor networks' Proceedings of IEEE International Conference on Advanced Information Networking and Applications (AINA), GinoWan, Okinawa, Japan, 410-417, 2008.

[2] I.F. Akyildiz, W. Su, , Y. Sankarasubramaniam, , and E. Cayirci, A survey on sensor networks. IEEE communication magazine. 40 (8) , 2002, 102-114.

[3] G. Pottie, and W. Kaiser, , Wireless IntegratedNetwork Nodes, Communication of the ACM. 43 (5). 2009, 51-58.

[4] L. Arboleda, and N. Nasser, , Comparison of Clustering Algorithms and Protocols for Wireless Sensor Networks. in Electrical and Computer Engineering, Canadian Conference on, Canada, 2006, 1787 - 1792

[5] S. Tilak , N. Abu-Ghazaleh, and W. Heinzelman, A taxonomy of wireless micro-sensor network models. ACM SIGMOBILE Mobile Computing and Communications Review. 6(2), 2002, 28 – 36.

[6] M. Sichitiu , Cross-layer scheduling for power efficiency in wireless sensor networks. in INFOCOM 2004. Twenty-third AnnualJointConference of the IEEE Computer and Communications Societies,(Hong Kong, Japan , 6, 2004, 1740 - 1750

[7] J. Ding, K. Sivalingamand R. Kashyapa, A multi- layered architecture and protocols for large- scale wireless sensor networks. in Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th., 2010,1443-1447

[8] Y. Fang, and B. McDonald, Dynamic Codeword Routing (DCR): A Cross-Layer Approach for Performance Enhancement of General Multi-hop Wireless Routing. in Sensor and Ad Hoc Communications and Networks (SECON). Boston, MA, USA, 2004, 255 – 263.

[9] H. Fei, A. Teredesai, and W. Hongyi, Timing- controlled, low-energy data query in wireless sensor networks: towards a cross-layer optimization

approach, in Networking, Sensing and control conference. Arizona, USA, 2005, 1031-1036.

[10] B. DeCleene, V. Firoiu, M. Dorsch, and S. Zabele, Cross-Layer Protocols for Energy-efficient Wireless Sensor networking,. in Military Communications Conference. New jersey, USA, 2005, 1102 – 1107.

[11] L. Zadorozhny, V., Chrysanthis, and P. Krishnamurthy, A Framework for extending the Synergy between MAC layer and Query Optimization in Sensor Networks, in the First International Workshop on Data Management for Sensor Networks. Toronto, Canada, 68-77, 2004.

[12] S. Kulkarni, A. Iyer, , and C. Rosenberg, An address- light, integrated MAC and routing protocol for wireless sensor networks. IEEE/ACM Transactions on Networking (TON), 14 (4). 793 - 806 2006.

[13] M. Mitzenmacher, and E. Upfal, Probability and computing. University of Cambridge, United Kingdom, 2005

[14] A. Mainwaring, , D. Culler, J. Polastre, R., Szewczyk, and J. Anderson, Wireless sensor networks for habitat monitoring. in the 1st ACM international workshop on Wireless sensor networks and applications. Atlanta, Georgia, USA, 88-97, 2002

[15] A. Sobeih, , W-P Chen, J. Hou, L. Kung , N. Li, H. Lim, , H. Tyan, and H. Zhang, J-Sim: A Simulation and Emulation Environment for Wireless Sensor Networks, IEEE Wireless Communications Magazine, 2006, 104 - 119

[16] P. Levis, N. Lee, M. Welsh, and D. Culler, TOSSIM: Accurate and scalable simulation of entire tinyOS applications. in the 1st ACM international conference on Embedded networked sensor systems. Los Angeles, USA, 2003, 126 - 137

[17] O. Younis, and S. Fahmy., Distributed Clustering for Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach. in INFOCOM. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, Hong Kong, Japan , 2004, 366 - 379

[18] B. Karp, Greedy Perimeter Stateless Routing for Wireless Networks. inMobiCom the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking ,Boston, MA, 2000.

[19] RFC3561, "Ad hoc On-demand Distance Vector(AODV) Routing," July 2003.