QoS-Oriented Asynchronous Clustering Protocol in Wireless Sensor Networks

Juan Luo, Hui-zhong Liu, Ren-fa Li
School of Computer and Communication
Hunan University, Changsha, China

Lichun Bao
Donald Bren School of ICS
University of California, Irvine, USA

Abstract — **The quality of service (QoS) to deliver time-and delay-sensitive data is critical in events-driven sensor network applications, which are constrained by energy consumption requirements. So far, most clustering proposals on this problem were based on periodic synchronous approaches, which require time synchronization and are not suitable for events-driven applications. We propose a QoS-oriented events-driven asynchronous clustering protocol, called EEAC (Energy-Efficient Asynchronous Clustering), which can deliver traffic in a timely and reliable manner. In EEAC, clustering starts asynchronously according to a probability, determined by cluster-heads' data transmission rate and residual energy. EEAC avoids time synchronization and adopts composite formula to elect cluster-heads. Simulation results show that EEAC ensures the real-time transmission of sensitive data, reduces the packet loss rate, and evenly distributes nodal energy consumption, thus prolonging network lifetime.**

Keywords – wireless sensor network (WSN); event-driven; clustering; quality of service (QoS).

I. INTRODUCTION

Wireless sensor networks (WNSs) consist of a large number of small low-power sensors that communicate through wireless links. One of the important applications of wireless sensor networks is to collect environmental data, as simple as temperature, humidity, light intensity collection [1]. On the other hand, multimedia contents, such as informative images, audio, video and other media are also increasingly being placed in WSNs for object monitoring purposes at a much finer and precise granularities [2][3]. The quality of services (QoSs) in WSN multimedia applications could also be significantly different in many respects. For instances, in military monitoring sensor networks, results of periodic measurement of a regional key parameter require reliable transmission, and low packet loss rate. While in the event-driven oriented enemy object recognition and tracking applications, the real-time data (such as streaming media) transmission is very sensitive to delay and jitter.

In order to satisfy the energy constraints and the QoS requirements for the WSNs, clustering has been a common and active approach to organize sensor networks into two tiers – cluster heads and cluster member. Cluster heads take charge of the establishment and maintenance of the cluster structure, collecting cluster members' data, and then send them to the sink node [4] after fusion processing. The cluster-based network models reduce time periods that nodes are active communicating with base station, and evenly distribute the energy consumption to all nodes so as to extend the sensor network lifetime. In this paper, sensor network lifetime is defined as the time from the start of network to the death time of the first node.

LEACH [5] clustering protocol proposed by Heinzelman *et al.* was the earliest and most representative of the low-power cluster routing protocols in sensor network. LEACH runs by round, and each round is divided into two stages of setup and stable operations. In order to improve the quality of elected cluster head, LEACH-C was proposed using a centralized clustering algorithm [6], in which the base station adopts simulated annealing algorithm to form optimal clusters so as to evenly distribute the network energy consumption. Younis *et al.* presented a new clustering model called HEED[7], in which cluster heads are elected through finite iteration, considering nodal residual energy and the inner cluster's communication costs. The quality of clustering in HEED is better than LEACH, but requires higher communication costs, and the time synchronization discrepancy is relatively large. DEEG [8] and HEEDM protocols are also based on such iteration way to choose cluster head.

The aforementioned research mainly focused on distributing energy consumption among cluster nodes, but the events-driven aspects issues were not considered. Most research so far assumed that all nodes collect and send data at the same rate and networks energy consumption is uniform, so that they regulate the run-time of each round, and periodically run the clustering algorithm in synchronous manner. However, in event-driven sensor network applications, events occur randomly and transiently, and accompanied by the bursts of large numbers of data, therefore, network energy consumption is uneven. The events-intensive region or time segment consumes more energy, while the events-sparse region or time segment consumes less energy. Hence, the synchronous clustering protocols are not suitable for event-driven based sensor network applications.

We propose an asynchronous clustering protocol in this paper, called EEAC (Energy-Efficient Asynchronous Clustering), for event-driven sensor networks. The asynchrony means cluster head can autonomously decide the clustering occasions according to a certain probability, rather than deterministically. This probability lies on cluster head's data transmission rates and residual energy. EEAC provides QoS guarantee and energy-efficient features.

II. EVENTS-DRIVEN QOS CHARACTERISTICS

The goal of event-driven sensor network applications is to detect event quickly when it occurs, and transfer such information promptly to the data centers. Three aspects of these applications need to be addressed. First, event detections

978-1-4244-2108-4/08/$25.00

need to be in real time, it is essential for network to perceive events as soon as possible. Secondly, the networks need to be ready quick enough to transfer the information. As events generate frequently and continuously, such as when tracking objects' situations, the networks needs to prepare proper number of nodes to reliably transfer these perceived event information to the data centers. The third requirement is to localize event impact only to the nodes involved in event detection and information transports so as to save energy in the space and time domains.

Most of the state-of-art clustering protocol design research shows some of deficiencies on event-driven sensor network applications in wireless sensor networks. First, energy consumption might be unevenly loaded to the sensor nodes in the space and time domain, and is unable to reflect the true network event state. Secondly, when the clustering algorithms run in periodic and synchronous fashion, the network may not be in a ready state to transport network traffic when the event happens, thus causing data loss and delays.

To resolve these issues, we propose an asynchronous clustering protocol, called EEAC (Energy-Efficient Asynchronous Clustering). EEAC operates asynchronously, in which clusters are formed according to its current residual energy and the traffic transport requirements. Therefore, it balances network energy consumption rate, and ensures the urgent data to be sent to the destination in an expedite manner. In addition, EEAC avoid time synchronization and constant topology maintenance efforts, thus saving network energy consumptions.

III. EVENTS-DRIVEN ASYNCHRONOUS CLUSTERING

We assume that a wireless sensor network (WSN) consists of homogeneous group of sensors with wireless communication capabilities. Wireless sensors are deployed in certain density such that the WSN is strongly connected overall. For convenience, we define the following terms in our discussions:

- Because sensors in WSNs are based on finite battery operations, we define the *network lifetime* as the moment from WSNs being deployed till the moment the first sensor runs out of battery power.
- *Clustering interval* is defined as the time between two consecutive cluster head generations.
- *Cluster traffic demand* is the amount of data to be sent to the WSN for transport to the data centers.
- *Energy threshold* for clustering is the remaining energy level at which the sensor nodes can no longer participate cluster head elections. In this paper, the energy threshold is denoted by E_1, and E_2 as the original energy.

The goals of EEAC are to find out the next round of cluster forming occasion, and to elect new cluster head members to form a new cluster-structure.

A. Clustering Occasion Detection

In event-driven sensor networks applications, the events detection brings along network traffic demands. Therefore, EEAC infers the event occasions according to the traffic demands v. The higher the traffic demands, the greater probability of event occasions. In addition to traffic demands, we consider the residual nodal energy as another factor for clustering in EEAC.

Therefore, we derive the clustering probability P of a cluster head for the next round as in Eq. (1).

$$P = \min\left(1, C_1 \cdot \frac{E_2 - E}{E_2} \cdot \exp\left(-C_2 \cdot \lambda^v \cdot \frac{E - E_1}{E_1}\right)\right). \quad (1)$$

in which, E_1 is the threshold energy to start clustering, E_2 is the original energy; v is the current cluster head's data transmission rate; E is the current nodal energy; C_1, C_2 and λ are constant. The value of P indicates the probability of the cluster head to start the next round of cluster head elections.

E_1 and E_2 ensure that cluster head has enough energy to carry out clustering functions without depleting its battery by excessive consumption. In Eq. (1), when $E = E_2$, P = 0, which means that the cluster head will keep the cluster head role. When the nodal energy E decreases, P increases; when the nodal energy $E \le E_1$, P = 1, which indicates that the node has to run the clustering algorithm to elect new cluster head as early as possible. That is, when $E \le E_1$, we require that $C_1 \cdot \frac{E_2 - E}{E_2} \ge 1$, which is equivalent to $C_1 \ge \frac{E_2 - E}{E_2}$.

The traffic demand v influences the time to choose a better clustering occasion. Given the same residual energy levels, the greater v is, the less P is, which means that when network is busier, the probability to run the clustering algorithm is lower, thereby ensuring the event-driven QoS requirements by running clustering algorithm only when network is in idle. Moreover, the effect of v on P is exponential with base λ ($\lambda > 1$), while E has linear impact on P. When traffic is heavy, we can see that the impact of v on P grows much stronger than that of the energy level E.

According to probability P, cluster head generates a random number r between 0 and 1. If r it is less than P, the cluster head participates the next round of clustering.

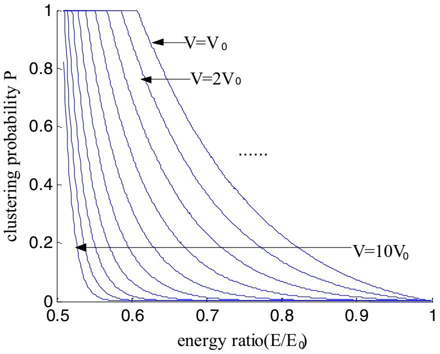

Figure 1. Clustering probability's trend with the change of data transmission rate and nodes energy.

Figure 1 illustrates the relationship between the clustering probability P and the other parameters when C_1 = 4, C_2 = 1.5, E_1 = 0.5E_0, E_2 = 0.9E_0, E_0 is the maximum energy level, v varies from v_0 to 10v_0, and v_0 is the basic cluster traffic demand.

We see that when the nodal energy E reduces from E_2 to E_1 and the traffic demand v is fixed, the clustering probability P rises from 0 to 1. When the traffic demand v is low ($v = v_0$), the curve climbs slowly, but when $v = 10v_0$, the curve is hardly affected by E when E is relevantly large and maintains more stable state, but when E is close to E2, the curve rises vertically and rapidly reaches 1. This explains that Eq. (1) well proportions network clustering occasion. In network busy state when v is great, the clustering occasion is postponed as long as possible; while in idle state, the network clustering probability P is higher under the same energy conditions. Therefore, this algorithm can reduce the network delay and data packet loss, and achieve requirements of events QoS.

B. *Cluster Head Election*

In the new cluster head election phase, each node in the cluster reports its current location and energy to the cluster head. Then, the cluster head calculates the average energy according to their reports, and the nodes whose energy levels are below the average energy are not chosen candidates for the cluster head. Afterward, the current cluster head selects a node with optimal location as the new cluster head. The optimal location means that the variance of distances between one node and all other member nodes is the lowest.

Although selecting the optimal location is an NP-problem, it is easy to solve this problem when the cluster size is small, which is usually about 10 nodes. In larger clusters, we can apply the simulated annealing algorithm to find out.

At last, the current cluster head broadcasts the newly elected cluster head's information to all the cluster members.

C. *Cluster Construction*

The cluster construction algorithm of EEAC is as follows:

```
Broadcast NEW_HEAD_MSG (NewCH,2*Radius)
On receiving a NEW_HEAD_MSG from $S_j$
if  $S_j$ = beSelfHead and NewCH = $S_i$  then
     Set  $E_1(S_i)$  and  $E_2(S_i)$
     Broadcast  I_Am_Head_Msg($S_i$)
elseif  $S_j$=beSelfHead and NewCH !=$S_i$  then
     IsCovered ← false
     Set_Wait_HeadMsg_Time(T)
elseif  $S_j$ != beSelfHead and $S_i \in S_{(CH)}$  then
     if  $E(t_{now}, S_i)$ >= $\theta$* $E(t_{init}, S_i)$  then
         Broadcast  I_Am_Head_Msg($S_i$)
    end if
end if
On receiving a I_Am_Head_Msg from $S_j$
if  IsCovered = false  then
    $S_{(CH\text{-}Set)} = S_{(CH\text{-}Set)} \cup \{ S_j \}$
end if
On expiring Set_Wait_HeadMsg_Time(T)
if  $S_{(CH\text{-}Set)} \neq \varnothing$  then
    IsCovered ← true
    beSelfHead = least_cost($S_{(CH\text{-}Set)}$)
else
    Set  $E_1(S_i)$  and  $E_2(S_i)$
      Broadcast  I_Am_Head_Msg ($S_i$)
end if
end
```

After cluster heads broadcast the information of newly elected cluster heads, nodes will select one cluster to join according to their own preference. First of all, the former cluster head s_j broadcasts the message of cluster dissolution with high transmission powers, which contains the new cluster head ID. The power level needs ensure neighbor cluster heads receive this message.

The neighbor node S_i which can receive the broadcast from S_j is divided into three categories:

(a) The newly elected cluster head. That is, $S_i = NewCH$. In this case, S_i calculates the current energy $E(t_{now},S_i)$, sets the critical energy $E_1(S_i)$ and $E_2(S_i)$ according to the Eq. (2) (3), and broadcasts it is a cluster head;

$$E_1(S_i) = \varepsilon_1 E(t_{now}, S_k) \quad (2)$$

$$E_2(S_i) = \varepsilon_2 E(t_{now}, S_k) \quad (3)$$

$$E(t_{now},S_k) >= \theta E(t_{init}, S_k) \quad (4)$$

in which, ε_1, ε_2 and θ are tunable parameters, influencing the algorithm performance. ε_1 determines the energy threshold E_1; and ε_2 determines the original energy level E_2. With regard to the values of ε_1 and ε_2, if ε_1 is low, it might lead to long clustering interval, which easily depletes the battery of the cluster head. If ε_2 is large, it might lead to short clustering interval, which easily causes instability and low energy efficiency of the network because of frequent clustering. θ determines the size of a cluster. The algorithm ensures the dynamic changes of network clustering structure by allowing nodes to join neighbor cluster, thus avoiding a fixed network clustering structure. At the same time, a reasonable θ can also adjust the size of a cluster to prevent the number of cluster heads from continuously increasing in the process of algorithm running, thereby better proportioning the energy consumption.

(b) The ordinary node in cluster. That is, S_j is the current cluster head of S_i. After receiving the broadcast, S_i sets itself to be in holdoff state with a waiting time for other cluster heads to broadcast message. the time is proportional to its current energy E;
An ordinary node waits for messages from newly elected cluster heads. If a node is in holdoff state, it will set this newly elected cluster head into its temporary cluster head set $S_{(CH\text{-}Set)}$. When its waiting time is over, if $S_{(CH\text{-}Set)}$ is non-empty, the node chooses a cluster head from $S_{(CH\text{-}Set)}$ with minimum cost to join; Otherwise, it broadcasts itself as a new cluster head, and sets the critical energy according to Eq. (2)(3).

(c) The neighbor cluster head. $S_{(CH)}$ is the set of all network cluster heads. S_i calculates its current energy $E(t_{now}, S_i)$. if meets the condition in Eq. (4), it broadcasts a message claiming itself as a cluster head, welcoming other nodes to join. In Eq. (4), $E(t_{init}, S_k)$ represents the energy of S_k when it is elected to be cluster head, and θ is constant.

IV. PERFORMANCE ANALYSIS

We used omnet++as the simulation platform to generate a network in 2000 by 2000 m^2 region in which 300 to 1000 nodes are deployed in random and static distribution. The initial energy is set to 10J, and the energy consumption of receiving 1 bit data is 50nJ. The signal propagation model is two-ray ground.

In the simulations, the time and locations of events occurred and their duration are random, and the duration of these events are also random, chosen between 50ms and 10s. The data transmission rate in the region of events is five times that of ordinary regions. We set other parameters, $\varepsilon 1 = 0.5$, $\varepsilon 2 = 0.9$ and $\theta = 0.75$.

Figure 2 shows the relations between the missing network events and the density of network nodes. In the simulations, we generated 1000 events randomly, and ran EEAC and other synchronous clustering algorithms LEACH, LEACH-C, and HEED for comparison purposes. Figure 2 shows that the event misses in EEAC are significantly lower than other synchronous clustering protocols; In addition, event loss of the network decreases when the network node density increases. This is because the increase in network density improves the chances of detecting the events and reporting to the data processing center of the network.

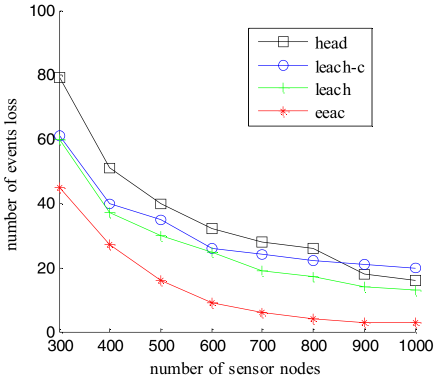

Figure 2. the relation between events loss and node density

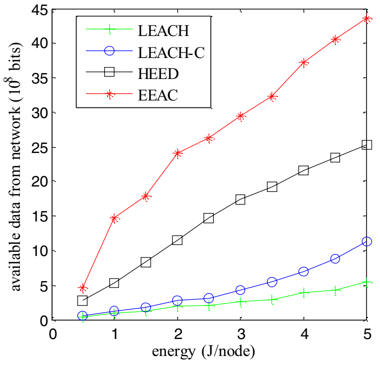

Figure 3. available data from network per unit energy

Figure 3 compares the amount of data provided from each unite energy in the aforementioned algorithms. Available data from network means the total data received by base station during network lifetime. In the simulations, the locality of nodal energy consumption is considered due to the events. The results indicate that asynchronous clustering protocol EEAC have higher energy efficiency than other three synchronous clustering protocols. Moreover, with the increase of initial energy, the increment of available data from network in EEAC is much higher than other synchronous clustering protocols. This is mainly because EEAC uses asynchronous clustering model mechanism, which can flexibly choose the appropriate clustering occasion according to its current network state. Therefore, it better proportions nodes energy consumption and improves energy efficiency. In addition, this is also because of the optimal locations of new cluster heads.

V. CONCLUSION

We have proposed EEAC, an event-driven QoS-oriented asynchronous clustering protocol for wireless sensor networks. In EEAC, cluster heads autonomously decides the clustering occasion according to its current data transmission rate and its residual energy level. The clustering process is asynchronous, rather than predetermined or unified managed by base station. The asynchronism of EEAC and the reasonable selection of clustering occasion effectively balance network energy consumption, and guarantee the reliable and real-time transmission of critical events.

ACKNOWLEDGEMENTS

This work is partially supported by the National Natural Science Foundation of China, Grant No.60673061, the National Research Foundation for the Doctoral Program of Higher Education of China, Grant No.2006053202, the National Science Foundation of Hunan Province of China under Grant No. 06JJ50111, 06JJ50113.

REFERENCES

[1] Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: A survey. Computer Networks, 2002,38(4):393−422.

[2] Cucchiara R. Multimedia surveillance systems. In: Aggarwal JK, Cucchiara R, Chang E, Wang YF, eds. Proc. of the ACM VSSN 2005. New York: ACM Press, 2005. 1−10.

[3] Tao D, Ma HD, Liu YH. Energy-Efficient cooperative image processing in video sensor networks. In: Ho Y-S, Kim HJ, eds. Proc.of the 2005 Pacific-Rim Conf. on Multimedia. Berlin: Springer-Verlag, 2005. 572−583.

[4] Krishnamachari B, Estrin D, Wicker S.The impact of data aggregation in wireless sensor networks Proceeding of the IEEE International Workshop on Distributed Event-Based Sytems(DEBS),Vienna, Austria,2002:575-578

[5] Heinzelman WR, Chendrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks Proceeding of the 33rd Annual Hawaii International Conference on System Sciences,Maui,HI, 2000:1-10

[6] Heinzelman WR. An application-specific protocol architecture for wireless microsensor networks. IEEE Trans. on Wireless Communications, 2002,1(4):660−670.

[7] Younis O, Fahmy S. Heed: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks. IEEE Trans. on Mobile Computing, 2004,3(4):660−669.

[8] Liu M,Gong HL,Mao-YC. A Distributed Energy-Efficient Data Gathering and Aggregation Protocol for Wireless Sensor Networks. Journal of SoftWare, 2005, 16(12); 2106-2116.