This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

Event-to-Sink Directed Clustering in Wireless Sensor Networks

Alper Bereketli   Ozgur B. Akan
Next generation Wireless Communications Laboratory (NWCL)
Department of Electrical and Electronics Engineering
Middle East Technical University, Ankara, Turkey, 06531
Email: {balper,akan@eee.metu.edu.tr}

Abstract— **Wireless sensor networks (WSN) are event-based systems based on the collaboration of several microsensor nodes. Due to the limited supply of energy at sensor nodes, energy-efficient configuration of WSN has become a major design goal to improve the lifetime of the network. Many clustering algorithms have been proposed as energy-efficient, however, the existing classical pre-event clustering solutions form clusters in the entire network unnecessarily that brings significant overheads in maintaining the network configuration. Unlike pre-event clustering, energy-efficient operation of WSN requires the *event-to-sink directed clustering* notion, which forms clusters when and where they are needed and in the direction of data flow from event location to the sink. To the best of our knowledge, energy-efficient clustering in WSN has not been studied from this perspective before. In this paper, we propose a novel Event-to-Sink Directed Clustering (ESDC) protocol for WSN. ESDC realizes energy efficiency in sensor network configuration by employing two techniques: (1) clustering of the nodes only within the event-to-sink data flow corridor to avoid unnecessary cluster formation, (2) directional clustering to minimize the number of hops for data forwarding. The directional clustering process in ESDC also sets up the routing path of the event flows over the clusters. Performance results reveal that the ESDC protocol achieves the energy-efficiency objectives and outperforms the existing conventional pre-event clustering approaches.**

I. INTRODUCTION

Wireless sensor networks (WSN) are event-based systems based on the collaboration of several microsensor nodes. The high density of sensor nodes is vital for sensing, intrusion detection, and tracking applications [1]. When an event is detected in the network, the aggregated collaborative report of the detecting nodes is delivered to the sink. To this end, clustering mechanisms enable the sensor nodes to collect and aggregate data at nodes called *cluster-heads* in each cluster, and this avoids redundant data flow in WSN. In this way, clustering removes some overhead in terms of packet generation, processing, energy consumption, and collision avoidance, because many functionalities of sensors are performed by cluster-heads. Hence, clustering helps prolong network lifetime and increases the scalability of WSN.

The main challenges in formation of energy-efficient clusters are the design criteria to select the cluster-heads, maintenance of them and integration of clustering approach with other mechanisms in the network such as coverage, data correlation and routing. In the current literature, a great deal of solution proposals exist for energy-efficient clustering [2]-[13]. Some of these studies offer pre-event clustering and routing solutions [2]-[6], [8], [10]-[12], while others present coverage-aware, energy-efficient clustering mechanisms [7], [9], [13]. *The main common drawback of these proposals is that they build clusters before an event occurs in the field or follow the pre-event clustering approach and form clusters in the entire field.* Intuitively, it brings significant overheads in terms of processing and energy.

On the other hand, event-driven clustering [14], [15] has not been fairly investigated in this domain. The event-driven clustering (EDC) protocol [14] exploits only the energy model of sensors and selects the cluster-heads according to the maximum remaining energy. Although the algorithm is named as event-driven clustering, the solution embeds some powerful heterogeneous nodes as gateways to forward the data packets from the event region to the sink node. Therefore, the performance of the protocol highly depends on the density and placement of such gateways, which may not be practical for many WSN scenarios. Furthermore, the data does not flow in the direction of sink rather toward gateway nodes that may require more transmissions and incur large delay. [15] presents an event-driven clustering paradigm for Wireless Sensor and Actor Networks (WSAN). In this framework, cluster-based routing protocol is applied only in the event region, where it forms a hierarchical tree of nodes with actor as the root of tree. In fact, it is a geographical routing rather than a clustering protocol without focusing on the clustering performance and does not discuss clustering procedure in detail. Moreover, this framework does not consider the event-to-sink data flow situation in WSN.

There are also some solutions proposed for clustering with routing based on the distance to the sink [16], [17], however, distance to the sink is not a sufficient criterion alone. Furthermore, the main emphasis in these papers is on energy conservation, aggregation, and routing. Consequently, for an energy-efficient operation of WSN, there is a need for a unified clustering solution which considers both event location and direction of data flow from event location to the sink. As a result, clustering must be *event-to-sink* and *directed*.

In this paper, we introduce Event-to-Sink Directed Clustering (ESDC) protocol that achieves energy-efficient clustering by employing two new techniques. *First*, *ESDC triggers*

This work was supported by the Turkish Scientific and Technical Research Council Career Award under grant 104E043.

978-1-4244-2948-6/09/$25.00 ©2009 IEEE

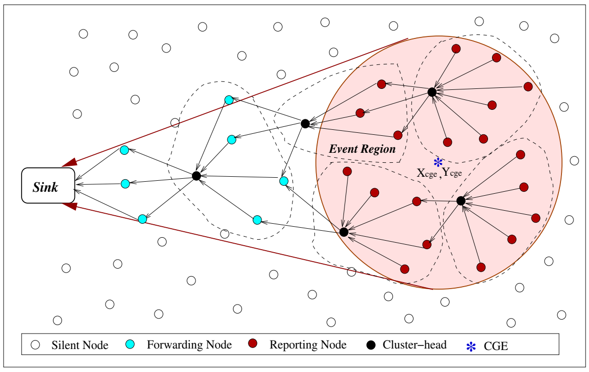


Fig. 1. Event-to-sink directed clustering approach of ESDC.

clustering only in the event region right after the detection of an event, and continues to form clusters in the corridor along the data forwarding path from event region toward the sink. Therefore, it avoids unnecessary clustering to prevent maintenance overheads. *Second, in order to minimize the number of transmissions in data routing, ESDC forms clusters such that among the group of neighbor nodes, the up-stream nodes are selected as the cluster-heads and down-stream nodes join only the upstream cluster-head.* That is, clusters are formed in the direction of event-to-sink that does not require to route the data back and forth in the cluster.

The remainder of the paper is organized as follows. The event-to-sink directed clustering notion is discussed and the operation of ESDC is described in detail in Section II. In particular, the operation of ESDC is best described in two parts, namely event-to-sink directed clustering and data path determination in ESDC running in parallel. Performance analysis and simulations of ESDC are explained and the results are given in Section III. Finally, the paper is concluded in Section IV.

II. EVENT-TO-SINK DIRECTED CLUSTERING

In this section, we describe the main idea of event-to-sink directed clustering approach and operations of ESDC protocol. ESDC aims to achieve energy-efficient clustering in two ways. First, clusters are formed only upon the occurrence of an event and only in the direction of data flow from the event region to the data sink without unnecessary cluster setup. Second, ESDC adapts directed clustering notion and chooses the cluster-head in the direction of the sink to minimize the delay of event flow. Thus, event flow is routed over the clusters that are formed along the shortest path from the event area to the sink.

A. Event-to-Sink Directed Clustering

In the event-to-sink directed clustering approach, ESDC forms the clusters only when it is required, that is, when an event occurs. The cluster formation process takes part in only the region in which event flow takes place, and does not go beyond that region to the whole network as illustrated in Fig 1. Here, we assume that most of the time events are just occurring in some parts of the field being monitored. Consequently, clusters are formed on the detection of an event and only in the direction of the minimum distance to the sink.

As an initial step, all nodes in the network periodically broadcast *Directed Clustering information (DCinf) beacons* within their local radio transmission range to maintain and update their neighbor lists. This DCinf beacon contains the following information:

- current energy E_i of the node i
- location of the node (x_i, y_i),
- transmission range (r_i) of the node,
- sensing range (s_i) of the node,
- a flag as a signal of event detection, $D \in [0, 1]$.
- weight of the node (ω_i).

Therefore, all nodes learn the locations of their neighbors. When an event occurs, nodes detecting the event multiplicatively increase their DCinf broadcast frequency f, i.e., $f = f.\phi$, and set the flag in their periodic DCinf beacon as an indication of the event. i.e., $D = 1$. Hence, nodes which do not take part in clustering and data routing are allowed to save energy by applying a specific duty cycle mechanism and entering sleep mode, since their broadcast frequency can be set to a much lower value than that of the nodes in the shortest event flow path. Moreover, the header of each data packet carries the following information to be used in directional ESDC clustering and data forwarding procedure:

- the location of the previous node from which a packet is received,
- the transmission range of the previous node from which the packet has arrived, and
- a flag indicating if the previous node was a cluster-head or not.

Nodes that receive the DCinf beacons sent by the sensors detecting the event calculate the *center of gravity of event (CGE)*. That is, the nodes estimate the center of the event region from the location information they have. Sensors receiving the DCinf beacons approximate the center of event region as the center of gravity found for the senders of the DCinf beacons, as in Fig. 1. For each node, the CGE is the center of gravity derived from the locations of the senders of the DCinf beacons. Those locations represent the nodes which have detected the event, and hence, their center of gravity approximately gives the location of the event. This approximation is valid in the cluster formation and data forwarding processes, since the objective is not localizing the event but determining the direction of clustering and routing. Hence, assuming that N nodes perform DCinf broadcast, sensors estimate the event location coordinates X_{cge} and Y_{cge} as the following

$$X_{cge} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad \text{and} \quad Y_{cge} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{1}$$

where x_i and y_i are the coordinates of a node detecting an event and broadcasting DCinf beacon. Each node i calculating the CGE from the DCinf beacons it received has a certain *weight* to become a cluster-head, i.e., ω_i. The weight (ω_i) depends inversely on the distance between the center of gravity (X_{cge}, Y_{cge}) and the node's location (x_i, y_i). That is, a node

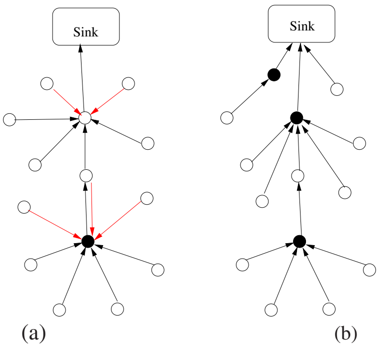


Fig. 2. (a) Undirected clustering, (b) Directed clustering and forwarding.

has a higher chance to become the cluster-head if it is closer to the CGE than its neighbors.

In the event region, cluster-heads are selected based on the value of the weights of the nodes. The weight of each node is obtained according to the following criteria:

1) A sensor has a higher chance of being a cluster-head if it is closer to the sink, because ESDC is designed to reach the sink using the shortest path to reduce energy consumption, delay and overhead. *Therefore, the weight of the node is inversely proportional to its distance to the data sink* (d_{sink}).
2) ESDC selects cluster-heads to be as close to the center of the event region as possible in order to make use of spatial correlation and efficient data aggregation. *The likelihood of becoming a cluster-head is inversely proportional to the node's distance to the CGE* ($d^i_{event} = \sqrt{(X_{cge} - x_i)^2 + (Y_{cge} - y_i)^2}$).
3) Data aggregation in the event region must be efficient in terms of energy, packet overhead, and number of clusters. Since there is some correlation between the data related with the same event detected by the sensors in the event region, data collected by the sensors that are spatially close to each other should not create different routing paths in the network. Therefore, a node in the event region has a higher probability of being a cluster-head if it can reach most or all of the sensors that have detected the event. *A node's weight is thus directly proportional to the number of different DCinf packets received form different event-detecting sensors* (n_{active}).
4) Obviously, a node that is going to be a cluster-head should have enough energy to aggregate data, to maintain its cluster, and to forward collected data toward the sink. *Therefore, the higher the remaining energy* (E_i) *of the node, the more likely it is to be a cluster-head.*

Consequently, the weight (ω_i) of the node i can be calculated as follows

$$\omega_i = \frac{c_1}{d_{sink}} + \frac{c_2}{d^i_{event}} + c_3 \frac{n_{inf}}{n_{nbrs}} + c_4 E_i \quad (2)$$

where c_1, c_2, c_3, and c_4 are some constant coefficients, n_{nbrs} is the total number of neighbors, n_{inf} is the number of nodes informing node i of an event by setting the event detection flag $D = 1$ in their DCinf beacons. Event data is collected and aggregated by the selected cluster-head, and it is again the responsibility of the cluster-head to send it in the direction of the sink.

B. Directed Cluster Formation and Route Establishment

ESDC algorithm prevents the cluster formation from spreading in the network, and clusters are formed within the event detection and flow region. Around the event region, after the clusters are formed with the DCinf beacons, the first cluster-heads are the ones collecting data from the event-detecting nodes. Data gathered by event-detecting nodes are sent to their corresponding cluster-head. The cluster-head forwards the packet to its neighbor that is closest to the sink. Continuously, each node on the way getting this first data packet forwards it to the neighbor having the smallest distance to the sink. *Shortest event flow path* is determined together with the forwarding of the first data packet in this way.

During the transport of the first data packet, the other process completed is the *directed cluster formation*, as illustrated in Fig. 2. In contrast to the undirected clustering shown in Fig. 2(a), data is not sent to a cluster-head back in the direction of the event. Instead, clusters are formed only in the direction towards the sink, and cluster-heads are selected on the shortest path to the sink, as in Fig. 2(b). In the header of the packet, there is an *ESDC flag indicating if the previous node was a cluster-head or not.* After the first cluster-head around the event region forwards the data packet to its neighbor that is closest to the sink, its neighbor checks the packet and sees that the previous node was a cluster-head. It chooses its neighbor that has *the smallest geographical distance to the sink* as the *next node*. It forwards the packet to the next node and selects it as the cluster-head. Hence, on the shortest event flow path, data is forwarded along the cluster-heads joined by a *forwarding* node in between. The *next node* has now received the packet from a *forwarding* node. It knows the location and transmission range of the cluster-head before the forwarding node from the DCinf broadcast, and it has learned that the sender of the packet was a forwarding node by looking at the packet header. It also knows the locations of its neighbors. Therefore, it checks its neighbor-list and selects the neighbors which are farther from the sink than itself and are in the transmission ranges of both itself and the *previous cluster-head*, so that they can be used as forwarding nodes in case of congestion. As the selected neighbors become the members of this cluster, the cluster-head informs them about this selection, and tells them to keep their communication units turned on until the end of the event.

This situation is illustrated in Fig. 3. Here, the shortest event flow path including nodes CH_1, C, and CH_2 is shown. CH_1 and CH_2 are two cluster-heads, and in the shortest path, C has taken part in as a forwarding node. CH_2 knows the location and communication range (r_{CH_1}) of CH_1 and locations of all its neighbors. For each neighbor node i, CH_2 checks if

$$d(i, CH_1) \leq r_{CH_1} \;\; \& \;\; d(i, sink) \geq d(CH_2, sink) \quad (3)$$

where $d(i, CH_1)$, $d(i, sink)$, and $d(CH_2, sink)$ are the distances between CH_1 and node i, between node i and the

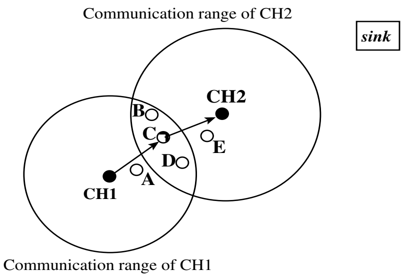


Fig. 3. Directed cluster formation along the event-to-sink corridor.

sink, and between CH_2 and the sink, respectively. Only the nodes satisfying this condition can link the cluster-heads, thus they are the nodes that can be the cluster members of CH_2 and the possible forwarding nodes between the two cluster-heads. Checking this, CH_2 chooses nodes B, C, and D as its members. Although nodes A and E are also in the data forwarding region, they cannot be chosen as forwarding nodes, because they cannot connect the two clusters directly, and increasing the number of forwarding nodes brings nothing other than unnecessary overhead in terms of energy and delay. Informed by CH_2, nodes B, C, and D know that they belong to the cluster of CH_2, and they keep their radio unit active all along the event communication period.

As a final note, it is worth discussing a multi-event case. During the directed cluster formation along the corridor from the event to the sink, the aim of each data receiving node is the transport of data to the sink along the shortest path. There is no difference between two event flow paths corresponding to two distinct events. Nodes do not check which event the carried data belongs to, and a node takes part in data transport for multiple events, as long as it is in the shortest event flow path for that event. Hence, in case multiple events are detected, the directed cluster formation allows the union of multiple event corridors at some cluster-head node.

The overall operation of ESDC protocol is also outlined in the pseudo-algorithm given in Fig. 4.

III. PERFORMANCE EVALUATION

In the literature, the main drawback of the energy-efficient clustering solutions [1]-[12] is that they build clusters before an event occurs in the field or follow the pre-event clustering approach and form clusters in the entire field. This approach results in significant overhead in terms of processing and energy. A well-known representative of such solutions is LEACH [7]. In our analysis, we evaluate the performance of ESDC and compare with the classical LEACH-based undirected pre-event clustering approach in terms of energy efficiency, packet delivery ratio, packet delay, and the effect of duty cycle.

```
    **ESDC()**
    /* Find the center of gravity of event
using the neighbors list */
    CGE(x, y) = FindCGE();
    /*Distance from the CGE */
    $Dist_{CGE}$ = Distance(My(x,y), CGE(x,y));
    /* Distance from sink node */
    $Dist_{Sink}$ = Distance(My(x,y), Sink(x,y));
    /* Number of neighbors i.e. either reporting
nodes or forwarding nodes */
    $Num_{Neighbors}$ = CountNbrs();
    /*Calculate my weight */
    $W_{my} = c_1/Dist_{CGE} + c_2/Dist_{Sink}$ +
       $c_3 \times Num_{Actives}/Num_{Nbrs} + c_4 \times E$
    /*Find the maximum weight of the neighbor
nodes which have not joined any cluster.*/
    $W_{NbrMax} = GetMaxIdleNbrsWeight()$;
    if (is-active $\&\& W_{my} > W_{NbrMax}$
       && my-status == NONE)
      my-status = CLUSTER-HEAD;
      /*Broadcast DCinf with cluster-head info */
      AnnounceHead();
    end;
    /* Packet received from neighbor node $i$ */
    when PACKET_RECEIVED
    if (pkt-type == DCinf)
       if (my-status == NONE &&
         Status(i) == CLUSTER-HEAD)
         if (Distance(i(x,y), Sink(x,y)) <
           Distance(My(x,y), Sink(x,y)))
           my-status = MEMBER;
           my-head = i;
        end;
      end;
    else /* Data packet */
       if (my-status == MEMBER)
        next = my-head;
      else
    /* Find the more recent nodes that has sent
DCinf packet and is more close to the sink */
         next = FindRecentFarthest();
      end;
       ForwardTo(pkt, next);
    end;
```

Fig. 4. Pseudo-algorithm of Event-to-Sink Directed Clustering.

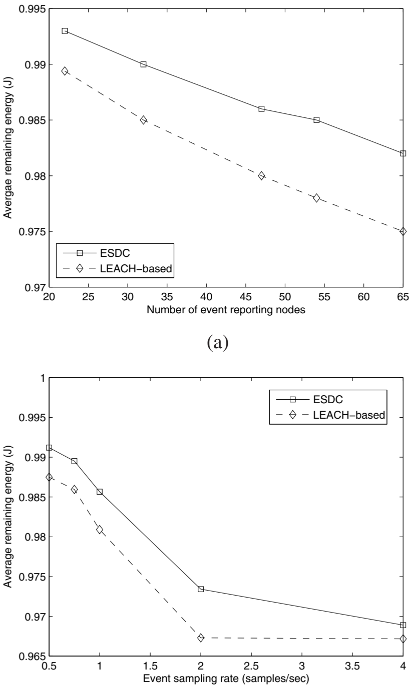
TABLE I
SIMULATION PARAMETERS

(c_1, c_2, c_3, c_4)	(1,5,1,1)
ϕ	15
Number of nodes	150
Field	100×100 m^2
Transmit power	100 mW
Sensing range	10 m
Initial node energy	1 J
DCinf beacon length	44 Bytes
Data packet length	100 Bytes
Active duration	2 sec
Passive duration	1 sec

The performance of ESDC is evaluated by using the Network Simulator *ns-2* [18] with the parameters summarized in Table I. The example scenario of wireless sensor network consists of 150 sensors deployed randomly in a field of 100×100 meters and sink is placed at $X = 0$ and $Y = 50$ in the field. An event source is also included in the scenario to trigger events with different event radius by using the NRL phenomenon node extensions [19] for *ns-2*. A sampling rate is the rate at which sensor nodes sample some event signal and generate data packets. Hence, by changing the event radius, the number of sensor nodes reporting event also changes.

A. Energy Consumption

The energy gain of event-to-sink directed clustering compared to pre-event clustering is shown in Fig. 5 (a). Here, LEACH-based pre-event clustering consumes 33% higher energy than ESDC. This is mainly because ESDC does not



(a)

(b)

Fig. 5. Average remaining energy for (a) event sampling rate of 1 sample/sec, (b) varying event sampling rate.

perform clustering in the entire network, instead, it forms clusters only within the event region as well as along the event flow corridor until the sink. Note also that the energy saving achieved by ESDC further increases with the number of reporting nodes as observed in Fig. 5 (a).

On the other hand, when nodes are sampling the event phenomenon at a higher sampling rate, more data is delivered to the sink node. Therefore, ESDC saves energy only by 20% as compared to LEACH-based pre-event clustering as given in Fig. 5 (b). It is due to the fact that ESDC delivers data at high rate under high traffic load as shown in Fig 6. For sampling rate greater than 1.5, ESDC almost maintains its delivery ratio and it is approximately twice as that of LEACH-based pre-event clustering. On the path from the event to the sink, cluster-heads are determined according to their remaining energies, number of received packets, and their distances to the event and to the sink. Cluster formation along the shortest flow path avoids possbile bottlenecks, and hence it is much more advantageous than single-path routing in terms of packet loss, delay and energy consumption. Moreover, in the case of multiple events, directed cluster formation distributes the traffic load over the nodes participating in cluster formation and routing, maintaining a certain degree of reliability and efficiency. Finally, since cluster-heads are selected and updated according to the weights of nodes, as shown in (2), cluster-heads do not cause any bottleneck in the data flow from the event region to the sink.

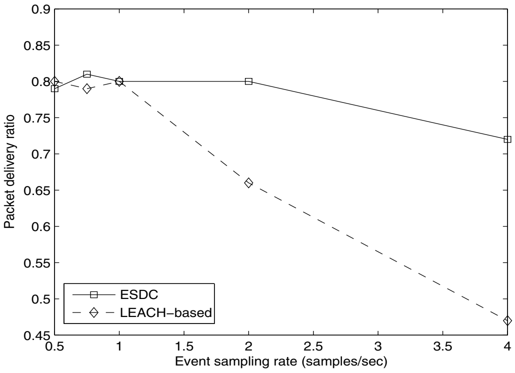


Fig. 6. Packet delivery ratio for varying event sampling rate.

B. Packet Delivery Delay

In undirected clustering mechanisms, member nodes always send their packets to heads, although some of them might be closer to the sink than their cluster-heads which may select the same node to forward the packet to the sink. This increases the route length, and hence, the packet delay. On the contrary, ESDC reduces the packet delay by employing the directed clustering and forwarding approach that minimizes the number of transmissions by forwarding data in the direction of event region to sink. Fig. 7 shows the performance of ESDC in terms of packet delay. It is clear that ESDC achieves 30% lower packet delay.

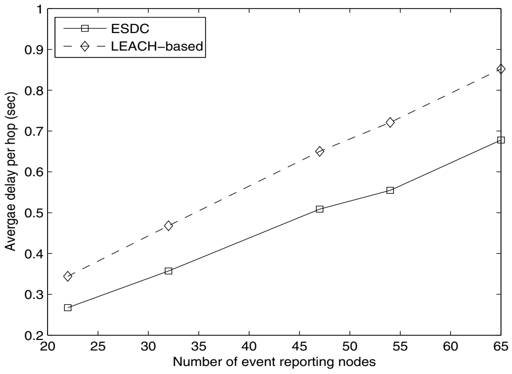


Fig. 7. Average packet delay for varying number of event reporting nodes.

We evaluate the performance of ESDC under varying sampling rates where the total number of reporting nodes are one third of the total nodes. As observed in Fig. 8, ESDC keeps the packet delay low at high sampling rate and per hop delay is approximately 200 ms even when the events are sampled at a rate of 4 samples/sec. However, this value is 460 ms in LEACH-based pre-event clustering that is greater than twice of ESDC. This low delay is achieved due to directed clustering and forwarding that minimizes the number of transmissions and intuitively minimizes the packet delay.

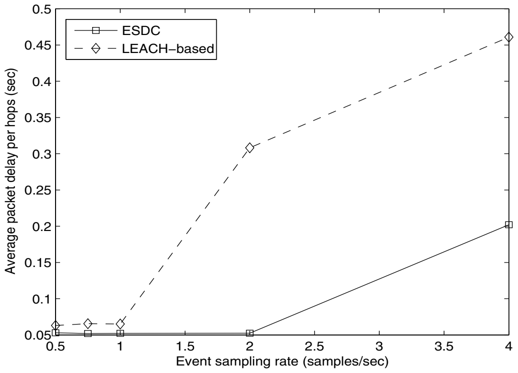


Fig. 8. Average packet delay for varying event sampling rate.

IV. CONCLUSION

In this paper, we present Event-to-Sink Directed Clustering (ESDC) protocol that achieves energy-efficient clustering in WSN. Unlike the existing conventional pre-event clustering solutions in the literature, energy-efficient operation of WSN requires to form clusters when and where they are needed and in the direction of data flow from event location to the sink. To the best of our knowledge, energy-efficient clustering in WSN has not been studied from this point of view before. Performance results show that the event-to-sink directed cluster formation and data routing approach of ESDC has the following advantages:

1) With the new *event-to-sink and directed clustering* notion, ESDC avoids the formation of unnecessary clusters in the network. Unnecessary cluster formation causes additional energy consumption, processing and maintenance overhead, and some possible delay in data forwarding.
2) Setting up clusters with respect to the event location prevents redundant nodes from taking part in cluster formation and data transport and avoids the extra energy loss and processing delay they would cause.
3) Event-to-sink cluster setup also utilizes the spatial correlation between the nodes in the event detection area. Since data from individual nodes are often correlated in a microsensor network, the data sink needs a higher-level description of the event occurring in the environment. Since ESDC considers the correlation between the sensor nodes in the event area, much less actual data has to be transmitted from the cluster to the sink. This way, energy dissipation and total overhead on the sensor nodes is reduced.
4) Directed clustering is accomplished by selecting cluster-heads to be the closest nodes to the sink. Hence, ESDC provides gain in energy, reduces packet transmission and processing overhead, and hence increases network lifetime.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, “A Survey on Sensor Networks”, *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102-114, Aug. 2002.
[2] S. Lee, J. Yoo, T. C. Chung, “Distance-Based Energy Efficient Clustering for Wireless Sensor Networks”, *in Proc. 29th Annual IEEE International Conference on Local Computer Networks (LCN)*, pp. 567-568, Nov. 2004.
[3] S. Bandyopadhyay, E.J. Coyle, “An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks”, *in Proc. 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) 2003*, vol. 3, pp. 1713-1723, Mar. 2003.
[4] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, “Next century challenges: scalable coordination in sensor networks”, *in Proc. the 5th annual ACM/IEEE international conference on Mobile computing and networking (ACM MobiCom 99)*, pp. 263-270, Aug. 1999.
[5] A. Manjeshwar, D. P. Agrawal, “TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks”, *in Proc. the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, pp. 2009-2015, Apr. 2001.
[6] A. Manjeshwar, D. P. Agrawal, “APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks”, *in Proc. the 16th IEEE International Parallel and Distributed Processing Symposium*, pp. 195-202, Apr. 2002.
[7] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks”, *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660-670, Oct. 2002.
[8] M. Younis, M. Youssef, K. Arisha, “Energy-Aware Routing in Cluster-Based Sensor Networks”, *in Proc. the 10th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS 02)*, p. 129, Oct. 2002.
[9] S. Ghiasi, A. Srivastava, X. Yang, M. Sarrafzadeh, “Optimal Energy Aware Clustering in Sensor Networks”, *Sensors*, vol. 2, no. 7, pp. 258-269, July 2002.
[10] K. Dasgupta, K. Kalpakis, P. Namjoshi, “An Efficient Clustering-based Heuristic for Data Gathering and Aggregation in Sensor Networks”, *in Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 3, pp. 1948-1953, Mar. 2003.
[11] O. Younis, S. Fahmy, “HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks”, *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366-379, Oct. 2004.
[12] G. A. Shah, O. B. Akan, M. Bozyigit, “Multi-Event Adaptive Clustering (MEAC) Protocol for Heterogeneous Wireless Sensor Networks”, *in Proc. The IFIP Fifth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2006)*, pp. 30-37, June 2006.
[13] J. Salzmann, R. Behnke, D. Lieckfeldt, D. Timmermann, “2-MASCLE - A Coverage Aware Clustering Algorithm with Self Healing Abilities”, *in Proc. 3rd International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP 2007)*, pp. 739-744, Dec. 2007.
[14] Z. Zeng-wei, W. Zhao-hui, L. Huai-zhong, “An Event-Driven Clustering Routing Algorithm for Wireless Sensor Networks”, *in Proc. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1802-1806, Sept. 2004.
[15] T. Melodia , D. Pompili , V. C. Gungor , I. F. Akyildiz, “A distributed coordination framework for wireless sensor and actor networks”, *in Proc. the 6th ACM international symposium on Mobile ad hoc networking and computing*, pp. 99-110, May 2005.
[16] Y. Xu, J. Heidemann, D. Estrin, “Geography-informed energy conservation for ad hoc routing”, *in Proc. the 7th annual ACM international conference on Mobile computing and networking (ACM MobiCom 01)*, pp. 70-84, July 2001.
[17] S. Lee, J. Yoo, T. Chung, “Distance-based Energy Efficient Clustering from Wireless Sensor Networks”, *in Proc. the 29th Annual IEEE International Conference on Local Computer Networks (LCN)*, pp. 567-568, Nov. 2004.
[18] UC Berkeley, LBL, USC/ISI and Xerox PARC, “The Network Simulator ns-2”, *The VINT Project, http://www.isi.edu/nsnam/ns/.*
[19] NRL’s Sensor Network extensions to NS-2, *http://downloads.pf.itd.nrl.navy.mil/archive/nrlsensorsim/.*