Distributed Asynchronous Clustering for Self-Organisation of Wireless Sensor Networks

Peter Hebden*[a], Adrian R. Pearce[a]

[a] *NICTA* Victoria Research Labortaory, Department of Computer Science and Software Engineering, University of Melbourne, Melbourne, Australia

This paper presents a fully distributed asynchronous clustering protocol for the self-organisation of a wireless sensor network into an infrastructure of well separated cluster heads that supports in-network processing, routing, and deployment. In this protocol, nodes volunteer asynchronously for cluster head duty and use a radio beacon to pre-emptively recruit members. Limited beacon range is used as the primary parameter for self-organisation. The resulting topology substantially reduces total transmission distance and the expected energy consumed by radio communication. To further extend network lifetime and capability, well separated cluster heads may be easily located and replaced by more powerful devices.

1. Introduction

In this paper we present a hierarchical clustering protocol designed for a large scale Wireless Sensor Network (WSN) of thousands of power constrained data sources, one base station ("sink"), and a data gathering application that continuously monitors the network for an aggregate value such as the mean temperature recorded by all sensors. Sensor network design may be influenced by many factors [1], and each device may differ in terms of its capabilities [2]. However, the fundamental challenge is to reduce the energy consumed by radio communication such that network lifetime is maximised given the application's quality of service (QoS) guarantees.

The Distributed Asynchronous Clustering (DAC) protocol provides an effective, low cost solution to an essential problem: how to generate a near optimal number of well separated cluster heads in a wireless deployment. DAC solves this problem by using local knowledge of the network and its environment to distribute cluster heads and terminate their generation. As a result, this protocol has many desirable properties. *Fully distributed:* Autonomous cluster heads use a limited range radio beacon to recruit members. By utilising local information, such as the relative strength of radio signals from nearby cluster heads, sensor nodes are able to make decisions that move network topology toward a global optimum. Such decentralised systems are more scalable and more robust against individual node or link failures [3]. *Adaptive:* The probability of a node volunteering for cluster head duty depends on its power level, its beacon range depends on network density, and the shape of each cluster depends on the radio propagation environment. *Self-terminating:* It stops after all nodes have decided to be either a cluster head or a cluster member. *Low overhead:* Clustering requires minimal communication of infrastructure data.

We implemented a simulator to model the impact of clustering on communication costs. In particular, we modeled the impact of clustering on transmission distance for the benchmark task. We assumed that a protocol that reduced distance more effectively in our simulator would also reduce the energy used by radio communication more effectively in a real deployment.

We compared the performance of DAC with another low overhead protocol, LEACH [4,5], and a hypothetical k-means [6,7] protocol. We found that DAC required fewer cluster heads to cluster all sensors, and generated a superior topology

---

*Corresponding author.
E-mail addresses: {phebden,adrian}@csse.unimelb.edu.au

where connectivity provided a far greater reduction in transmission distance.

The primary advantages/characteristics of this protocol are: (1) *pre-emptive recruitment* promotes well separated cluster heads, (2) the number of cluster heads can be optimised, (3) it provides a solution that is fully distributed, scalable, adaptive, and robust, (4) facilitates deployment of devices - especially over irregular terrain, and (5) provides substantial performance advantages over existing protocols.

In Section 2 we discuss clustering for self-organisation and in-network processing. In Section 3 we present results from our simulations. In Section 4 we present related work. And finally, our conclusions.

2. Hierarchical Clustering for In-network Processing and Routing

A WSN is a *complex system* composed of a large number of nodes and their interactions and, consequently, many researchers have taken a hierarchical clustering approach to self-organising sensor networks. Hierarchical architectures may take advantage of device heterogeneity and create opportunities to exploit the computation - communication trade-off. For the energy cost of transmitting 1Kb a distance of 100 meters, a general purpose processor with 100MIPS/W power could execute ≈ 3 million instructions [8]. As the scale of a deployment increases, it becomes more important for network topology to use hubs and provide opportunities for in-network processing.

2.1. Hierarchical Clustering

For large scale deployments, where fine grained sensing covers a wide area, networks will include tens of thousands or even millions of nodes. “Protocols will have to be inherently distributed, involving localized communication, and sensor networks must utilize hierarchical architectures in order to provide such scalability.” [9] Indeed, hierarchical clustering and in-network processing are well known and scalable techniques for reducing communication costs. They may reduce transmission distance and the amount of data transmitted. In general, sensor nodes are partitioned into a set of clusters where each sensor may only transmit data to its cluster head. Cluster heads process member data and forward the result to the base station. However, existing clustering protocols tend to suffer from limitations related to latency, overhead, centralisation, global knowledge of the network, lack of scalability, and suboptimal numbers of poorly distributed cluster heads.

2.2. In-network Processing

One of the key motivations for clustering protocols such as DAC is to facilitate in-network processing, i.e. perform computation, such as data aggregation, close to the data source such that the amount of data transmitted is reduced. Further, some applications will require that network nodes process data cooperatively and combine data from multiple sensors. Transmitting raw data from all sensors to a base station does not scale well and because per node throughput scales as $\frac{1}{\sqrt{N}}$ - it goes to zero as the number of nodes N increases [10]. In other words, as N increases every node spends more time forwarding packets of other nodes.

In the next section we provide a summary of LEACH [4,5] before presenting DAC because it highlights several important problems and is most comparable to our protocol.

2.3. LEACH Clustering Protocol

Low-energy adaptive clustering hierarchy (LEACH) is an application specific protocol architecture for homogeneous WSNs [4,5]. To develop their clustering protocol they assumed that all nodes are powerful enough to transmit directly to the base station when necessary. Their network self-organises into clusters using a semi-distributed algorithm. Individual nodes use global knowledge of the network to make autonomous decisions about whether or not to volunteer for cluster head duty and which cluster to join without any centralised control. The probabilities of volunteering during set-up are intended to generate an optimal number of cluster heads.

At the beginning of self-organisation, nodes may autonomously volunteer for the role of cluster head. The probability P_i of sensor S_i volunteering depends on its energy level and whether or not it has volunteered in any of the previous (r *mod* (N/C)) rounds. The expected number of cluster heads C depends on N and the set of probabilities P at time t [5]. However, the probability P_i does not depend the decisions made recently by other sensors in the network.

After all nodes have decided their role, cluster heads broadcast an advertisement message (beacon) to the WSN using a Carrier Sense Multiple Access (CSMA) protocol. The range of this beacon is **not limited**. Free sensors associate with the cluster head that requires minimum communication energy, i.e. the closest based on *received signal strength*.

LEACH clusters tend to be highly variable in terms of the number of clusters, cluster size, cluster head distribution, and percent of sensors clustered. Figure 2 shows the most obvious problem: cluster heads are not well separated. Given these problems, LEACH-centralised (LEACH-C) was proposed in [5].

2.4. DAC Clustering Protocol

The design of the DAC protocol is based on two observations. (1) A cluster head's beacon range and the number of clusters formed are very closely related. (2) Multiple proximate nodes should not volunteer. However, most protocols rely primarily on residual energy to influence the probability of a node volunteering for cluster head duty, ignore decisions made by proximate nodes, rely to some degree on global knowledge of the network, and fail to adapt to the environment.

DAC relies primarily on the strength of a cluster head's beacon and *pre-emptive recruiting* rather than the probability of a sensor volunteering, to generate a near optimal set of *well separated* cluster heads. Under *pre-emptive recruiting*, when a sensor volunteers for cluster head duty, it transmits a limited range beacon and recruits neighbors as soon as possible to prevent them from volunteering. In contrast, we would say that LEACH uses *non pre-emptive recruiting*. These differences enable DAC to consistently outperform LEACH without incurring more overhead.

As the pseudo code in Figure 1 shows, during the DAC set-up phase, free sensors volunteer to be cluster heads with a certain probability, each new cluster head emits a beacon according to 1-persistent CSMA, free sensors within beacon range are "recruited" and set themselves to *not free*, record the received signal strength (RSS) and identity of each beacon heard. Given enough time, every sensor will hear a beacon or volunteer. After a waiting period, during which no more beacons are heard, each sensor associates with the nearest cluster head based on RSS.

Fault tolerance may be provided by a variation of the DAC set-up phase. When a cluster head fails, the sensors associated with it will gradually become aware that their cluster head no longer transmits and change their status from clustered to free. After a waiting period, a qualified free sensor will volunteer to be a cluster head, broadcast a beacon, and free sensors within range will associate with it. Over time the network will degrade gracefully, reporting less detailed information about its environment.

```
while (sensor.isFree == true)
{
  if (sensor.random(0..1) < p)
      sensor.isFree = false
      sensor.isClusterHead = true
      sensor.setBeaconRange( )
      sensor.broadcastBeacon( )
  else if (sensor.hearsBeacon( ))
      sensor.isFree = false
      sensor.recordRSS( )
}
sensor.listen(time_period)
clusterHead = sensor.maxRSS( )
sensor.associate(clusterHead)
```

Figure 1. Pseudo code for each node during set-up phase of DAC.

2.5. Deployment

DAC facilitates deployment of devices in several ways. First, a large number of low cost homogeneous sensors may be deployed in an approximate manner over an irregular terrain. Clusters should adapt naturally to the radio propagation contours of the environment. Second, fault tolerance reduces the need for maintenance and deployment of new devices. Third, some applications will require greater in-network processing and/or longevity. We can use the sensor network itself to facilitate an upgrade. Well separated cluster heads, which are relatively few in number, could be automatically located and replaced by more powerful devices to facilitate small-world properties [12].

3. Simulation and Results

We compared the performance of LEACH, DAC, and a hypothetical k-means clustering protocol. The most important metric was the *total transmission distance* required for our benchmark task after clustering. In our simulations, each protocol was subjected to the same constraints. These included a limited cluster head beacon range and a limited number of cluster heads C.

We also investigated the relative importance of the following network parameters: radio model, base station location, and beacon range. However, we tried to keep our model as simple as possible [13]. Our simulations were intended to present a simplified representation of reality - just enough detail to unambiguously reveal the relative impact of three clustering protocols on total transmission distance.

3.1. Network Model

In most simulations the network was composed of 875 sensors deployed approximately 30 units apart on a grid of 25 rows and 35 columns. Each sensor's x, y coordinates were perturbed slightly by a small Gaussian. The base station was located at $x = -50$, $y = -50$. Cluster head radio beacon range was limited to a Euclidean distance d of 150 units (distance is relative). The path loss exponent n depended on assumptions about distance and the environment.

Radio communication is highly variable and difficult to model [5]. However, since quantitative estimates of packet loss and expected network lifetime for a deployment of specific hardware are outside the scope of this paper, we made a number of simplifying assumptions. To compare protocols, each node was assumed capable of direct communication: sensor node to cluster head, cluster head to base station and, in the case of unclustered sensors, sensor to base station.

3.2. Radio Model

The strength of a transmitted signal decreases in proportion to d^n, where d is distance, and n is the path loss exponent. Depending on multi-path and other interference, n is typically in the range of 2 to 5 [3]. The actual value strongly depends on the radio propagation environment [14].

We assumed a simple radio model where the radio consumes a fixed amount of energy per bit to run the transmitter or receiver circuitry, and a variable amount per bit to run the transmit amplifier. In LEACH [4], the variable amount is proportional to transmission distance d^2. In their later work [5], the variable amount is different for intra-cluster communication and base station communication, d^2 and d^4 respectively. However, in both [4] and [5], it was found that approximately 5% of sensors should be cluster heads. If communication with the base station consumes energy proportional to d^4 instead of d^2, then this should reduce the optimal number of cluster heads.

So we tested LEACH using different radio models and unlimited beacon range. For d^1 path loss, the optimal percent of cluster heads was $\approx 5\%$. For d^2 path loss, the optimal percent of cluster heads was $\approx 2.5\%$. However, in other simulations where we assumed a higher path loss exponent for cluster head to base station communication, such as the d^2, d^4 model used in [5], **one cluster head** was optimal, i.e. one cluster head minimised clustered network distance. So, it is not possible to generate useful results where unlimited beacon range is assumed.

These preliminary tests suggested several implications. The optimal number of cluster heads is not necessarily $\approx 5\%$, rather it depends on the radio model, path loss exponents in particular, and the beacon range. Given a limited beacon range, the optimal number of cluster heads should be inferred by the network itself.

3.3. Number of Cluster Heads and Coverage

The desired number of cluster heads C must be known *a priori* for both LEACH and k-means in order to calculate the probability of a sensor volunteering. In contrast, DAC relies primarily on beacon range R in its attempt to generate an optimal number of clusters - it does not require *a priori* knowledge of the network. This difference means that DAC stops generating new clusters automatically. In contrast, LEACH can easily generate too few or far too many clusters because its sensors do not use local information, i.e. whether or not a nearby sensor has already volunteered. If too few cluster heads are generated, then some sensors will not be clustered ("orphans") because they are out of beacon range. If too many, then channel contention is increased unnecessarily.

All else being equal, a smaller number of cluster heads is preferred because limited wireless channel bandwidth must be shared by all sensors and cluster heads in the network. After clusters have been formed, each cluster head creates a TDMA schedule which tells its members when they can transmit. To reduce inter-cluster interference, each cluster communicates using direct-sequence spread spectrum (DSSS), also known as direct sequence code division multiple access (DS-CDMA), where each cluster communicates using a different CDMA code. All nodes will receive better communication channels when there are fewer clusters [5].

As Figure 4 shows, DAC consistently clustered more sensors with a limited number of cluster heads than LEACH. DAC is likely to cluster nearly 100% of sensors with only 28 cluster heads, substantially better than LEACH's 88%. LEACH required at least 60 cluster heads to cluster nearly all sensors consistently because it elects cluster heads at random locations - clustering the last few sensors required a disproportionate number of volunteers in order to get them where needed.

3.4. Cluster Quality

In our simulated network, an ideal set of clusters would have a uniformly and evenly distributed set of cluster heads. Figure 2 shows a representative set of poorly separated LEACH cluster heads. The contiguous gray patches contain nodes that are out of beacon range. Many cluster heads (large white circles) are much too close together. Figure 3 shows evenly distributed DAC cluster heads where all sensors are within cluster head beacon range. We compared the distance between each cluster head and the nearest adjacent cluster head for each protocol. The minimum distance between a DAC cluster head and its closest neighbor cluster head was consistently larger.

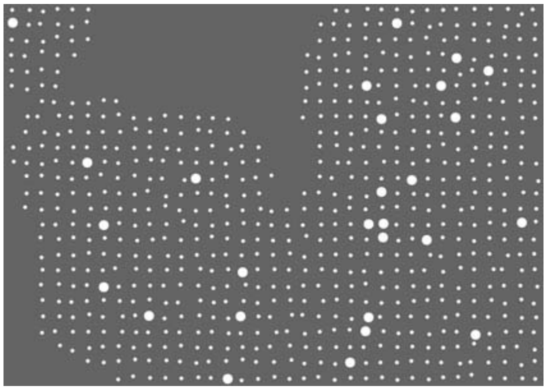


Figure 2. LEACH Clusters: 27 randomly distibuted cluster heads, 82% clustered.

3.5. Metrics

We simulated a *data gathering network* where all nodes sense their environment at a steady rate and report their data periodically. The benchmark task was to determine the average sensor measurement. For this task, the communication cost of delivering data to the base station dominates. Since the energy consumed by radio communication can be approximated by a function of

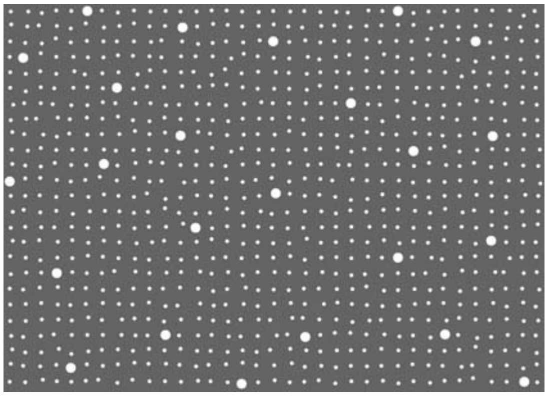


Figure 3. DAC Clusters: 24 evenly distributed cluster heads, 100% clustered.

Euclidean distance d, we used d^n as a metric for evaluating clustering protocols.

We defined *clustered network distance* (CND) as the sum of d^n distances: each clustered sensor to its cluster head, each unclustered sensor to the base station, and each cluster head to the base station. *Maximum network distance* (MND) was defined as the case where each of the N sensors must transmit raw data directly to the base station : $\sum_{i=1}^{N} d_i^n$. For each protocol we calculated clustered network distance as a percent of maximum network distance: $p_{max} = 100 * CND/MND$. As an indication of relative expected energy consumption we used the ratio of $LEACH.p_{max}/DAC.p_{max}$.

3.6. Clustered Network Distance and Radio Model

An estimate of energy consumption depends on the radio model, transmission distance, and the path loss exponent n in particular. So we ran simulations given $C = 28$ and different values for n to determine its relative impact on the clustering protocols.

First we assumed that all communications were over similar distances under similar conditions. As n increased, the expected relative energy consumption of LEACH versus DAC increased. As shown in Figure 5, from $n = 1$ to $n = 5$, DAC was 1.9, 3.9, 4.8, 4.9, and 5.5 times better respectively. Second we assumed that intra-cluster communications were over relatively short distances and under ideal conditions, d^2 power loss, and base station communications were over relatively long distances and subject to interference, d^4 power loss. According to this model, and given our parameters, a DAC network would be expected to last ≈ 5 times longer than LEACH.

3.7. Clustered Network Distance

The clusters generated by each protocol are intended to reduce communication costs by reducing the total transmission distance for our benchmark task, which is to compute the average sensor measurement at the base station. If a sensor is associated with a cluster head, then it only needs to transmit a short distance to its cluster head - where its measurement will be aggregated with other measurements and forwarded to the base station. Our metric for comparing protocols in this context was clustered network distance expressed as a percent of maximum network distance.

As Figure 6 shows, DAC reduced clustered network distance by a greater amount than LEACH - especially in the $2-4\%$ cluster heads range. Assuming d^2 path loss, clustered network distance decreased rapidly as the number of cluster heads increased from zero to 28 or $\approx 3\%$ of sensors, Figure 6. DAC was expected to cluster all sensors with about 3% cluster heads, and since *no sensors were free to volunteer at that point*, the number of cluster heads was limited. However, according to LEACH all sensors decide whether or not to volunteer first, which means any number could decide to volunteer.

Although a hypothetical k-means clustering protocol would require far more communication overhead than DAC, it did not perform better in our simulations. Assuming d^2 path loss, network distance after clustering, as a percent of maximum network distance, was 4.17% for DAC, better than the 5.05% for k-means.

Furthermore, even when LEACH was allowed to generate an optimal number of cluster heads according to its protocol, more than twice as many cluster heads as for DAC, it was still less effective. Based on 100 trials and d^2 path loss, LEACH required 7.4% cluster heads to cluster 98.61% of sensors and reduced network distance to 9.29%. DAC required about 3.1% cluster heads to reduce clustered network distance to 4.19% , still much better than LEACH, Figure 6.

Table 1 provides a summary of the clustering protocols and their relative effectiveness.

3.8. Optimal Number of Cluster Heads

An optimal number of well separated cluster heads should minimise clustered network distance. So, for each protocol we recorded the average number of cluster heads that minimised clustered network distance based on fifty trials. We observed that the number of cluster heads that minimised clustered network distance depended on several network parameters. For example, we tested LEACH and DAC with different values for the radio model, base station location, and beacon range.

For LEACH, given the d^2 radio model, an average 64 cluster heads was optimal, but when the d^2, d^4 model was used, 50 cluster heads was optimal (a 22% reduction). This suggests that when the cost of communication with the base station is greater than intra-cluster communication, less cluster heads are needed. However, this parameter did not have a significant impact on DAC, where we observed 31 and 30 cluster heads respectively.

Next we moved the base station further away from the sensor network, from (-50,-50) to (-500,-500). For LEACH, 56 cluster heads was optimal. The decrease was limited because the protocol needed enough cluster heads to cluster almost all sensors, which depends on beacon range. For DAC, 29 cluster heads was optimal.

And finally, we increased beacon range from 150 to 300 units. For LEACH, 22 cluster heads was optimal (a 66% reduction). For DAC, 9 cluster heads was optimal (a 71% reduction). While the radio model and base station location have an impact on the optimal number of cluster heads, beacon range is clearly the parameter that should be given the most attention.

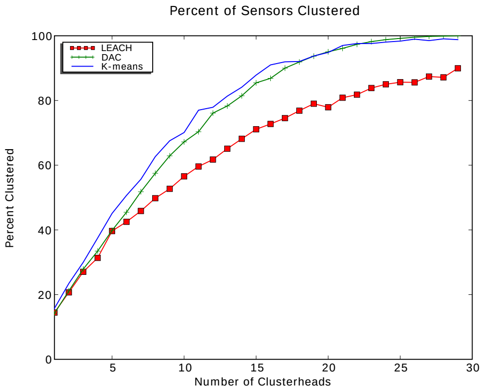


Figure 4. Percent of sensors clustered by three protocols. DAC clusters 100% of sensors with less cluster heads than LEACH and k-means.

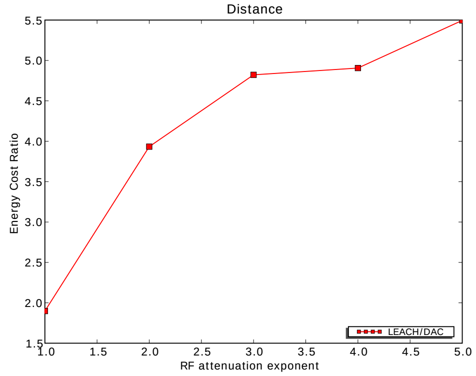


Figure 5. Expected energy consumption ratio: as the path loss exponent n increases, the relative energy consumption of LEACH versus DAC is expected to increase.

3.9. Scalability

We simulated networks of 200 to 14,000 sensors, using a beacon range of 150 units, and a d^2 radio model. As network size increased, the percent

Protocol	distributed	separation	distance
LEACH	no	poor	poor
DAC	yes	excellent	excellent
k-means	no	good	good

Table 1
A comparison of clustering protocols during each *round* of operation. DAC is fully distributed, and exhibits better CH separation and distance reduction.

of cluster heads required for DAC to cluster all sensors decreased slightly. In a small network of 200 sensors, we needed about 4% cluster heads, in a network of 875, 3.07%, and in a large network of 14,000 sensors, 2.85%. This reduction is not possible where the number of cluster heads is an explicit parameter.

Next we tested networks of 875 to 14,000 sensors. As network size was increased from 875 to 14000, LEACH's clustered network distance grew from ≈ 4 times greater to ≈ 5.5 than DAC's. DAC also performed better than a hypothetical k-means protocol.

3.10. Optimal Beacon Range

To determine the optimal beacon range, DAC was tested using beacon ranges from 10 to 500 in 10 unit increments and d^2 path loss. As shown in Figure 7, clustered network distance was minimised when beacon range was set at 210 units, which generated 15 cluster heads.

Then we compared LEACH with DAC where both used a beacon range of 210 units, and LEACH also used the explicit parameter of 15 cluster heads. DAC was still ≈ 4.2 times better at reducing clustered network distance.

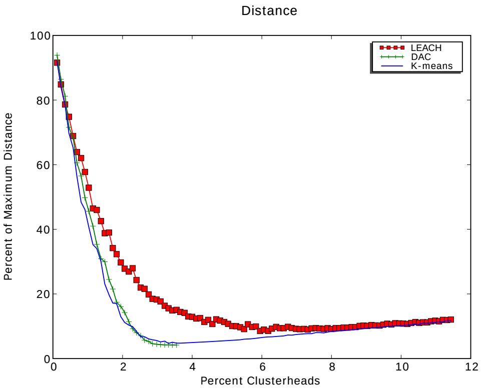


Figure 6. Minimising clustered network distance: Given a 150 beacon range and d^2 radio attenuation, DAC required ≈ 3% cluster heads, LEACH ≈ 6%.

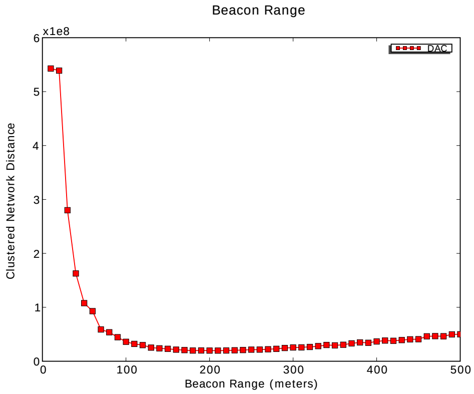


Figure 7. Optimal beacon range for DAC: 210 units, generated 15 cluster heads, ≈ 1.7% of sensors.

4. Related Clustering Protocols

Many research efforts have addressed the problem of reducing communication costs in data gathering networks. By optimising trade-offs network lifetime can be increased while still meeting a specific application's quality of service guarantees.

Threshold sensitive Energy Efficient sensor

Network protocol (TEEN) [15] and Adaptive Periodic TEEN (APTEEN) [16] were designed for *reactive networks*. In such networks, nodes are expected to respond immediately to sudden changes. Hard and soft thresholds were used to reduce message transmission. Naturally, this reduced the amount of data transmitted and extended network lifetime *vis-a-vis* LEACH, which is better suited for applications that require information based on periodic data from all nodes.

Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [17] was designed to be near optimal in terms of energy cost for data gathering applications. In their protocol, chain(s) of communication are built from neighboring nodes. When the authors compared protocols, they let PEGASIS use CSMA and forced LEACH to use TDMA. We note that to make a valid comparison, both protocols should be tested using the same MAC layer protocol. In summary, PEGASIS assumes that all nodes have global knowledge of the network, and energy savings come with a penalty of increased delay.

Bandyopadhyay and Coyle [18] proposed a fast, randomised, distributed algorithm for organising mobile sensors into a hierarchy of clusters. Their algorithm was designed to generate one or more levels of stable clusters with an optimal number of cluster heads on each level.

Distributed Energy-efficient Clustering Hierarchy Protocol (DECHP) [19] outperformed LEACH [5] in simulations partly because LEACH cluster heads communicate directly with the base station, while DECHP used multi-hop routing. Energy consumption was assumed to be proportional to distance d^4, so their simulation did not make a fair comparison. LEACH and multi-hop are not mutually exclusive, so a better comparison would require that both protocols use the same form of routing. While DECHP is an interesting protocol that addresses some of the problems associated with LEACH and other clustering protocols, its setup phase requires a large amount of communication overhead.

The Hybrid Energy-Efficient Distributed clustering protocol (HEED) presents fully distributed approach to promoting network lifetime [20] where each node uses local information to generate well separated cluster heads and elects cluster heads based on their residual energy.

5. Conclusion

We have presented DAC, a low overhead, energy efficient approach to distributed asynchronous clustering. In order to make straighforward and fair comparisons with other protocols, we assumed direct communication between nodes, and one level hierarchical clusters. Multi-hop routing and n level hierarchical clusters may provide additional energy savings and, depending on the application, many other optimisations are possible and compatible with DAC.

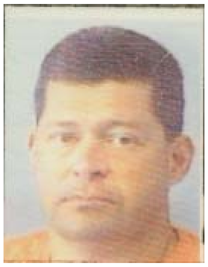
Based on our simulations, we observed that the number of cluster heads that minimses transmission cost depends on a number of parameters including the radio model and base station location, but cluster head beacon range had by far the greatest impact. We noted that because limited wireless channel bandwidth must be shared by all clusters, a protocol should self-organise the network such that clustered network distance is minimised with as few cluster heads as possible. In light of these observations, similar protocols may need about twice as many cluster heads as DAC where our standard network parameters, the d^2 radio model and a limited beacon range are assumed.

DAC uses a range limited cluster head beacon and pre-emptive recruitment to generate well separated cluster heads; it automatically stops generating cluster heads after all sensors have been recruited. This not only reduces communication costs, but also promotes connectivity via cluster heads, i.e. sensors should not be "orphans" when high capacity nodes are within range. Well positioned cluster heads may be replaced by more powerful devices to facilitate small world properties.

The benefits of DAC increased with the scale of the network, and with the cost of radio communication. DAC builds an infrastructure for in-network processing and routing via cluster heads [21] that extends the lifetime of large scale networks.

REFERENCES

1. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, A survey on sensor networks, *IEEE Communications Magazine*, August 2002.
2. K. Romer and F. Mattern. The Design of Wireless Sensor Networks, *IEEE Wireless Communications*, December 2004
3. Feng Zhao and Leonidas Guibas. *Wireless Sensor Networks: an information processing approach*, Elsevier Inc., 2004
4. W. Heinzelman, A. Chandrakasan and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks, *IEEE HICSS*, January 2000.
5. W. Heinzelman, A. Chandrakasan and H. Balakrishnan. An Application-Specific Protocol Architecture for Wireless Microsensor Networks, *IEEE Trans. Wireless Communication*, 1(4):660-670, Oct. 2002
6. A.K. Jain, M.N. Murty and P.J. Flynn. Data Clustering: A Review, *ACM Comp. Surv*, 1999.
7. R.O. Duda, P.E. Hart, & D.G. Stork, *Pattern Classification*, Second Edition., New York, NY: John Wiley and Sons, 2001.
8. G. J. Pottie and W. J. Kaiser. Wireless Integrated Network Sensors, *Communications of the ACM*, vol. 43(5), May 2000, pp. 51-58.
9. Bhaskar Krishnamachari. Networking Wireless Sensors, Cambridge University Press, 2005.
10. P. Gupta and P. R. Kumar. The capacity of wireless networks, *IEEE Trans. on Information Theory*, 46(2):388-404, 2000.
11. A. Scaglione and S. D. Servetto. On the interdependence of routing and data compression in multi-hop sensor networks, *8th Annual International Conference on Mobile Computing and Networks (MobiCom)*, Atlanta, GA, Sept. 2002.
12. A. Helmy. Small Worlds in Wireless Networks, *IEEE Communications Letters*, 7(10), Oct 2003.
13. Hal R. Varian. How to Build an Economic Model in your Spare Time, *Passion and Craft: Economists at Work*, Ed. M. Szenberg, University of Michigan Press, 1997.
14. Andreas F. Molisch. *Wireless Communications*, Wiley, 2005.
15. A. Manjeshwar, and D. P. Agrawal, TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks, *1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, San Francisco, CA, April 23-27, 2001.
16. A. Manjeshwar and D. P. Agrawal, APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks, *2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, April 15-19, 2002.
17. S. Lindsey, C. Raghavendra, and K. M. Sivalingam. Data Gathering Algorithms in Sensor Networks Using Energy Metrics, *IEEE Trans. Parallel and Distrib. Sys.*, vol. 13, no. 9, Sept. 2002, pp. 924 935
18. S. Bandyopadhyay, E. Coyle, Minimizing communication costs in hierarchically-clustered networks of wireless sensors, *Computer Networks*, 44(1): 1-16 (2004)
19. O. Moussaoui and M. Naimi. A Distributed Energy Aware Routing Protocol for Wireless Sensor Networks, *The 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, Montreal, Quebec, Oct. 2005
20. O. Younis and S. Fahmy. HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks, *IEEE Transactions on Mobile Computing*, Vol. 3, No. 4, Oct - Dec 2004.
21. P. Hebden and A.R. Pearce. Data-Centric Routing using Bloom Filters in Wireless Sensor Networks, *Fourth International Conference on Intelligent Sensing and Information Processing (ICISIP)*, Bangalore, India, December 2006.



Peter Hebden received a BA in Economics from the University of California at Berkeley and a JD in Law from the University of Utah. He is currently a PhD candidate in the Computer Science and Software Engineering at the University of Melbourne.

Adrian R. Pearce received a BS in Compter Science from the University of Melbourne and a PhD in Computer Science from the Curtin Institute of Technology. He is currently a Senior Lecturer in the Computer Science and Software Engineering at the University of Melbourne.