Exploiting Multi-Channel Clustering for Power Efficiency in Sensor Networks

Ashima Gupta, Chao Gui and Prasant Mohapatra
Department of Computer Science
University of California, Davis
Davis, CA 95616
{guptaa, guic, prasant}@cs.ucdavis.edu

Abstract--Sensor networks typically comprise of a number of inexpensive small devices with processing, communication and sensing abilities that collaborate to perform a common task. Sensor devices use batteries as their sole power supply. The operational lifetime of a sensor network, therefore, depends entirely on the better utilization of the devices. Typically a sensor network is divided into clusters to optimize power utilization by performing division of labor and data aggregation within a cluster. This paper introduces a novel approach to naturally distributed clustering of sensor nodes in a sensor net using multi channel data planes. Our technique incorporates a virtual sense mechanism that reduces energy spent in sampling and transmission. It also decreases network traffic, thereby decreasing contention, potential collisions and retransmissions. This approach inherently implements a sleep-awake mechanism based on virtual sensing that contributes towards increasing the network lifetime by efficient utilization. The proposed technique can be used to track spreading phenomenon like forest fires and water flows. A spreading phenomenon can be represented by a field whose value changes dynamically with time over area. We focus on following the movement of such a dynamically changing field rather than obtaining the value of the field at different locations at disjoint random times.

I. INTRODUCTION:

Sensor networks have been used for a number of applications ranging from the relatively static ones like environment monitoring to highly dynamic target tracking. Such networks can be used for monitoring a spreading phenomenon like forest fires, water flow etc. The sensor field for this application can be static for long periods of time but once the phenomenon starts and spreads, the sensor network will track the movement of the dynamic field. Hence this application models a medium range of dynamism where the static environmental monitoring and target tracking form each end of the spectrum of applications on the dynamic scale. This technique can be applied to the extreme scenarios but not optimized for them.

We define a dynamically changing field as one which changes its value distribution over an area varying with respect to time. The area in this case is limited by the field covered by sensor nodes. To further divide the problem, we define hot and cold zones. A hot zone is a generic term for the dynamic part of the sensor field where events of interest are taking place, and a cold zone defines part(s) of sensor field which has remained relatively static, i.e., has shown no change in data sampled over a predefined long period of time.

We track the spreading phenomenon by dividing the sensor field into hot and cold zones and then monitoring changes in the zonal areas as the zones merge or drift. By accurately determining and reflecting these changes we can follow the movement of the spread.

Sensor networks comprise of a large number of sensor nodes that coordinate to perform a specific task typically monitoring and surveillance. In most of the applications for sensor networks the field to be monitored will be static for majority of the time until the phenomena starts and spreads. When this happens the sensors will be actively sampling and transmitting data. Hence network traffic in a sensor network is very bursty and highly correlated. Bursty traffic will result in collisions and retransmissions causing higher energy expenditure and increasing the latency. Various clustering techniques are used to minimize this effect. Sensors are grouped into clusters and clusterheads become the point of data aggregation and compression. Clusterheads forward the aggregated data of their respective clusters to the sink. This decreases the amount of data being transmitted and relayed, as well as decreasing the contention. Hence clustering techniques provide for significant energy savings. A sensor field may be divided into clusters in a number of different ways e.g. grid [12]. Within a cluster, nodes may have sleep-awake and sample schedules based on the observation that data is highly correlated and latency is of secondary importance, hence all nodes within a cluster need not be active all the time. The clusterheads may either be nodes with higher capacity for power and processing (heterogeneous) or the same as any other node in its cluster (homogenous). In case of homogenous clustering, the clusterhead functionality has to be rotated among other nodes in the cluster in order to balance the load. We assume homogenous sensor nodes in the sensor network because of ease of deployment over large geographical areas like forests. The cluster formation may be centrally coordinated or done in a distributed manner. Given that the sensor field is fairly large and inaccessible after deployment, distributed clustering is preferred for its flexibility, scalability and fault tolerance. Finally, inter cluster and intra cluster *media access* is the critical enabling technology for sensor networks. The MAC may be TDMA or CSMA. Even if we consider stationary sensors in the sensor field, there still may be topological changes as sensors die and/or new sensors are added to the field. In such a scenario it would be difficult to scale a TDMA based scheme. We, therefore, work with a CSMA based MAC.

In this paper we focus on optimizing energy efficiency in the data sensing and reporting phase of the sensor network monitoring an environmental phenomenon that has a spreading or moving property. We ignore the static time period when the sensor network has not detected any reportable change in the environment. We note that such an application implies that nodes with high data correlation could be grouped since they represent a specific uniform field value. Therefore, all nodes that sample data and the data value is within a predefined range should form a cluster in a distributed manner. To achieve this distributed clustering without spending energy and bandwidth in nodes communicating with each other to find which nodes belong to which cluster we have derived a mechanism wherein each node would naturally know and communicate with its cluster mates only. Our approach is analogous to having a large group of people speaking various languages, then people who speak the same language and are within hearing range of each other form a group. Our scheme minimizes data transmissions and hence contention and energy expenditure.

The rest of the paper is organized as follows. The overall perspective of the approach is discussed in section II. The deployment of the sensors followed by cluster formation and communication is described in Section III-A. In Section III-C, we explain power efficiency and load balancing mechanisms used by the protocol. Section III-D describes how sensors will change planes to monitor the phenomenon based on their sampled data. Our simulation scenario and results are discussed in Section IV. We summarize the paper in Section-V and end the paper by suggesting possible areas of future research.

II. OVERALL PERSPECTIVE

Nodes sampling data periodically (the period is configurable) will communicate on a pre-specified channel that is different from other nodes. Therefore, if any one node within this cluster is transmitting data pertaining to an event then, other nodes within the same cluster that sense that the channel is busy will go to sleep, similar to the scheme described in [1]. However, the key difference is that the node will discard the data. It need not report the event at all, as it is already being done by a buddy node [2]. This virtual sensing mechanism can be used to answer queries as well. A queried node will sense the channel and if the channel is busy it need not sample for data and yet answer the query.

This technique increases the network lifetime by conserving energy. The energy consumed in the sensor networks can be factored into: transmit/receive energy, sensing, computation, overhearing and idle listening. Furthermore, in sensor networks communication is bursty in nature. Hence during a data burst additional power is consumed because of collisions, subsequent retransmissions and contention (backing off and sensing repeatedly). Bandwidth, therefore, becomes a bottleneck as a result of high network traffic. The multi channel scheme reduces overall network traffic as well as traffic on any individual channel. This reduces collision and contention thus reducing the energy expended. MAC Protocols described in [3], [4], [5] and more recently [6] optimize sleep-awake patterns of nodes in a sensor network to prolong network lifetime without causing a significant increase in the latency. These protocols can be used in conjunction with the multi channel sensing scheme described to obtain even better power efficiency. Our technique optimizes power utilization in a snapshot of time when an event occurs and multiple nodes have the same or similar data to report. Instead of transmitting redundant data and then wasting compute cycles of the clusterhead to compute aggregates, transmit duplicates or receive and discard duplicates, we eliminate it at the cluster member sensor level. This saves the transmission energy of the node, computational energy of the clusterhead and does not clog the bandwidth. However, since our scheme removes data redundancy at the sensor level, there could be some loss of accuracy if packets get dropped due to lossy media. Here we assume that the media is radio waves. Using the virtual sense mechanism, nodes may not even have to expend energy for sampling at times.

The technique described in [7] also uses unequal clustering to achieve power efficiency in sensor networks.

However, the key idea in this work is to balance the energy expenditure across sensors to maintain connectivity in the network over a long period of time. It is noted that sensors expend energy unequally. Sensors close to the base station have to act as relay for other nodes and clusterheads, hence they burn up energy and die sooner then other nodes. This results in a disconnected and effectively dead network. The authors observe that the amount of power used by a clusterhead depends on both inter as well as intra cluster communication. They try to reduce intra cluster communication for nodes with high inter cluster communication potential in order to balance the power usage across the network and attain a better network lifetime A layered network is used, where each layer is composed of a number of clusters that are same in size and shape but different from those in another layer. Layers closer to the base station will have smaller clusters. Our technique also uses unequal clustering but we deal with optimizing intra cluster communication.

A hierarchical clustering approach using heterogeneous sensors is discussed by Hempel et al in [13]. They focus on obtaining power efficiency using data aggregation and data bundling for watershed monitoring application. In contrast, we implement only two levels of hierarchy and use homogenous sensors and we do not aggregate but discard redundant data. The Data funneling approach described in [14] uses data compression and aggregation to decrease the number of transmitted packets and increase channel utilization by minimizing control overhead. This is based on the observation that most sensors reporting data at the same time would use similar header, hence compressing and clubbing them into one packet would decrease contention. Our approach on the other hand just discards the correlated packets as defined by the user configurable granularity level. Hence we do not require intra-cluster aggregation, though inter-cluster aggregation may be applied. There may be a slight loss of accuracy in our scheme; however it is not significant especially because we apply the technique to follow the movement of a phenomenon instead of the field value at a particular time. The data funneling approach also assumes that controller nodes have greater computational and communication abilities. However we prefer a homogenous composition of the sensor network primarily because of the ease of deployment. In [12] the sensor network is divided into grids based on routing equivalence of nodes within the same grid. Only one node per grid remains active while others are in sleep mode conserving energy. However our clustering criteria are based on data correlation, not routing equivalence. Power usage optimization techniques for the static monitoring phase of the sensor network are essential for ensuring longevity of the network lifetime. One such technique is discussed in [11] which use sparse tree topology to minimize the number of active nodes within the network while maintaining connectivity. We note that such techniques are orthogonal to our approach.

III. DETAILS OF THE APPROACH

A. *Deployment and Clustering:*

We assume a random and dense deployment of sensor nodes in the sensor field. In the example of a forest fire, sensor nodes will be aerially dropped over the forest (sensor field) with a more or less uniform distribution. We also assume that once deployed, each node will know its location and that all nodes are synchronized (Techniques for localization and time synchronization are not within the scope of this study). The nodes will remain static and communicate over RF media. After deployment the sensors have to form a network in order to maintain connectivity with the sink and to perform collaborative sensing. The usual method to do so is by forming clusters of sensor nodes. Some approaches to sensor node clustering are described in [8] and [9]. The clusterheads send cluster aggregated data to the sink. These schemes reduce network traffic to the sink and thus save transmission energy and also economize on the use of available bandwidth. The clusterheads form a backbone to maintain connectivity over the sensor field and with the sink. For this application we choose to group nodes based on their data correlation on the stipulation that geographically close nodes would typically have high data correlation. This would further enable us to have a direct mapping between zones and clusters.

We propose a multi channel clustering mechanism. The communication channel is logically divided into sub-channels based on the desired granularity of data. As mentioned earlier, geographical proximity implies high data correlation. Therefore, nodes that sample data, for example temperature between 10-20 degrees are likely to be closer to nodes that sample temperature within 20-30 degrees and their communication on successive channels may result in high interference. Hence, channel assignment should be done in a manner to minimize interference For example if we want to sense temperature 10 degrees apart then the channel may be divided into 12 sub-channels, where temperature below 10 degrees is reported on channel 1, between 10 and 20 degrees is reported on channel 11, between 20-30 degrees on channel 2, 30-40 degrees on channel 10 and so forth. Temperature over 100 degrees is reporter on channel 6. One channel say, Channel 0 is reserved as a common control channel. The non linear mapping as described above is one way to mitigate the effect of interference.

Nodes that sample data within a particular range will communicate over a predetermined channel. Hence all nodes on the same data plane will tune their radios to the same channel. The sensors will be clustered naturally by the channels they communicate on. This scheme has further savings in terms of power efficiency and virtual sensing as described in section III-C. The channel subdivision could be achieved in various ways e.g. using Code Division Multiple Access (CDMA). Efficient channel subdivision is not the focus of our paper and we will assume that an efficient scheme for channel subdivision would be used.

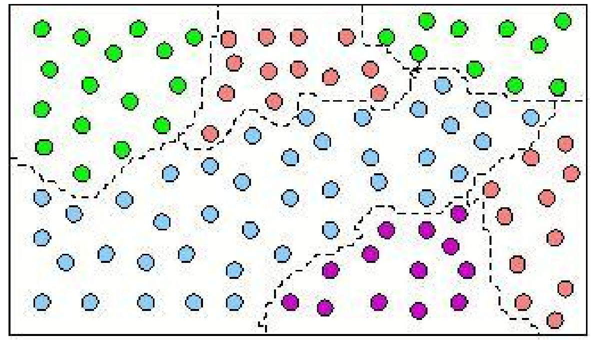
All sensor nodes have a default granularity setting that may be tuned by the user via the sink after the initial setup. Scalability for multi modal sensing could be achieved by decreasing the granularity level of the data planes. To make this technique scalable for finer levels of granularity, we can divide the data planes into more clusters and enable channel reuse across the clusters. Channel reuse will cause node synchronization issues. The consequential issues of node synchronization and channel reuse would be a topic for future research and are not discussed in this paper.

B. *Initial Setup:*

The sensors upon deployment sample data, and based on the default granularity setting tune their radios on the appropriate channel. All sensor nodes on the same data plane elect a clusterhead. Clusterhead election can be performed by using any of the methods described in [8] or [9]. Note that one data plane may be divided into more than one cluster depending on the connectivity of the plane based on the transmission range of the nodes, as shown in Figure 1.

Hence there is a one to many mapping between a data plane and clusters. In the worst case we could have a one node cluster, if it is not within the transmission range of other nodes within the same data plane. However, our stipulation of dense deployment precludes such a scenario unless all the other nodes in the cluster die or are damaged during deployment. The clusterheads communicate with each other and the sink on the control channel and thus form backbone of connectivity to the sink.

We note that each cluster can be mapped onto a geographical 'patch'. In the setup phase every sensor node will send its coordinates, value of data sampled and time stamp to its clusterhead on the data channel. The clusterhead will convey the aggregated information along with the data plane to the sink over the control channel. Initially the entire field is static; hence all the patches are cold zones.



Data Plane 1 corresponding to temp <= 40 degrees
Data Plane 2 corresponding to temp <= 20 degrees
Data Plane 3 corresponding to temp >= 80 degrees
Data Plane 4 corresponding to temp <= 60 degrees

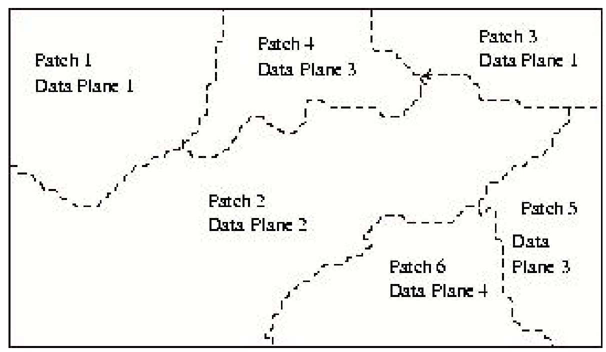


Fig 1. Data Planes and correlation with Patches/Clusters

C. *Power efficiency:*

We observe that in any particular data plane all the nodes are reporting more or less the same data from the user's perspective. Otherwise, the user would have tuned the planes for finer granularity. The redundant data does not provide any extra information but causes an energy drain, high contention and loss of bandwidth. Ideally, only one representative node should sample data at any given time while other nodes in the plane can sleep to conserve energy. Once the nodes have decided on their data planes, each node wakes up periodically to sample data and check if it is in the same data plane. There will be at least one active node (the clusterhead) per cluster in order to ensure connectivity and to allow nodes changing cluster to be able to announce its new membership. The active node will switch duties with an inactive node after a certain period of time. Hence while a node samples data and transmits sampled information, other nodes in the cluster that sense the channel is busy will go to sleep since data representing the cluster is already being transmitted. Therefore, nodes can virtually sense if their data is already being sent. If the occurrence of an event is sensed by a few nodes, there will be contention among the nodes for the channel in order to deliver the same event notification to the clusterhead. The first node that wins the contention starts transmitting the information. Other nodes sense that the channel is busy and discard the event notification information and go to sleep under the assumption that this data is already being transmitted by another node in the vicinity. Thus the virtual sense mechanism results in significant energy savings as shown in our simulations. This mechanism also results in fewer transmissions decreasing network traffic and contention, thus promoting optimal utilization of bandwidth. Typically a sensor node samples data either periodically in order to discover any plane change, or in response to a query. Sensor nodes using virtual sensing may not have to sample data on a query. If a node receives a query for data, then before sampling, it checks if the corresponding data channel is busy. If the channel is busy then the node can answer the query with a positive or negative response without having to sample data. Note that, this virtual sensing mechanism is a direct result of the multi-channel data plane clustering technique that we have described earlier.

A data plane may cover a large area. Therefore, to maintain connectivity across the corresponding cluster we propose to use a tree structure of active nodes rooted at the clusterhead as shown in Figure-2. Ideally the tree should represent a minimum cover for the cluster. The active nodes forming the tree are located such that every node in the cluster is within the transmission range of at least one active node.

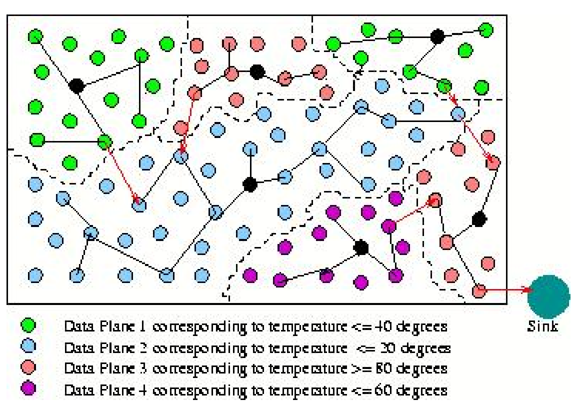


Figure 2. Spanning Tree within each cluster for connectivity to sink

1) Duty cycle of member nodes: Active nodes will change duties with passive nodes so that their battery is not completely drained similar to the concept described in LEACH [10]. However, the node selection process to takeover the tree, member functionality is not random. Unlike LEACH, we adopt a power aware selection process. Each node keeps track of the amount of work it has done in terms of transmissions (no of times and distance), sampling and receptions. Each time a node wakes up to check whether it is in the same data plane, it sends a message to the clusterhead via the nearest node in the tree. The message will contain its coordinates and amount of work it has done. Based on this information, it may be chosen to replace the nearest active node if its energy level is estimated to be more than that node. The energy level may be estimated by the amount of work that a node has done as described above or on the residual battery life indicated by the hardware. This technique would ensure a balanced load distribution among the nodes.

2) Duty cycle of clusterheads: The clusterhead functionality will also have to change duties with other nodes in order to preserve its energy. This can be done in the same manner as described above for tree member nodes. An active node within a defined range that is estimated to have the maximum residual power can be chosen to be the next clusterhead. Hence the choice of the successor to a clusterhead depends on two factors a) its distance to the current clusterhead and b) the energy spent by the node. Such a scheme would imply that centrally located nodes would be clusterheads more often than other nodes resulting in energy drain in this region. This could be a limiting factor for network connectivity and hence the network lifetime. This scenario would be probable if the plane remains static for a long period of time. If, however, a phenomenon causes a change in the shape of the plane, then the center of the plane would change accordingly.

D. Introducing Dynamism:

We now introduce some dynamism in the static sensor field and describe the protocol that is used to represent the changes. Every node periodically samples data to check if it is in the same data plane. If a node discovers that it has to switch planes, then it will advertise its arrival on the new channel by transmitting a broadcast message containing its coordinates and its transmission range (for extensibility to heterogeneous nodes). An active node that receives this transmission will acknowledge the reception and convey the message to the clusterhead. The clusterhead sends this information to the sink via the clusterhead backbone on the control channel. The clusterhead may send this information once for every node that joins its cluster or buffer the information and sends it as a periodic burst depending on the accuracy requirements of the application. The sink will designate this area as a potential hot zone. If more than a configurable number of nodes transition into this cluster, the corresponding geographical area is designated as a hot zone. A clusterhead only reports addition of nodes in its planes and not deletions. However, once the information of addition of nodes to a specific cluster reaches the sink, it can modify the global view of the sensor field that it has and relay precise information to the user. If multiple nodes transition to a data plane, they will contend for the channel to broadcast their arrival. Note that clusterheads may also have to contend for the control channel if more than one has data to send to the sink.

Nodes that change their data plane would also increase their sampling cycle based on the heuristic that if they detect a change once, they will detect more changes as the phenomenon spreads in intensity as well as geographical region. Once it reaches a stable state, that is, a configurable number (default is 5 in our simulation) of samples yield the same data value, the sampling cycle is again reduced. The clusterhead may broadcast beacons to all active nodes to increase their sample rate in order to detect subsequent changes. All passive nodes that wake up to sample data will also receive this message and update their sample periods. This is based on the heuristic that if a change has taken place in the direction of this cluster, then the phenomenon may be spreading in the same direction. Increasing the sampling rate would increase the accuracy in capturing the change in terms of detecting and reporting the change and subsequent plane changes. Furthermore, the sample rate may be increased in proportion to the number of nodes transitioning to this plane. If multiple nodes transition into a data plane, then they will contend for the data channel to update the clusterhead. As more nodes join a data plane, the data plane is designated as a hot zone by the sink and the changes to its area are reflected in the sink with every node that succeeds in obtaining the channel and advertising its arrival. We note that if a large number of nodes switch to a data plane in a small period of time the sample rate of the plane increases proportionally. Hence we get a finer level of data accuracy with the arrival of each node in the plane. Since with the increase in sample rate nodes within this plane may further move to a different plane as the phenomenon moves. For example if a forest fire is spreading, then as the temperature in an area hits 200 degrees and nodes in this region transition to the 200 degree plane, we can say a fire has started. As the fire spreads, more number of nodes that are actively sampling will detect the change sooner and hence be able to convey it to the sink with lower latency. In our current simulations, we just increase the sample period of the node changing plane to a configurable high sample period and on steady state to a configurable low sample period.

If a clusterhead transitions into another plane, it has to elect another head before it transitions. The easiest way to elect would be to choose one of its children to be the clusterhead, since all nodes on the tree are always active. This approach will incur a lower latency, but will not ensure connectivity with other children nodes of the clusterhead. The ideal way to perform clusterhead handover would be to choose the nearest active node that is the most centrally located within the cluster. This choice of the new clusterhead ensures connectivity in the tree structure while minimizing latency of communication from other nodes in the cluster. This approach would also decrease clusterhead handovers because of plane change as the clusterhead functionality would keep moving towards the center of the cluster so it will be less likely to make a transition as compared to the border nodes.

If a node transitions to a plane where there is no other node, it will detect the change by broadcasting its arrival for a fixed number of times and listening to the channel for acknowledgment for a time period T after each transmission before timing out to retransmit. If the channel remains free, it will self appoint itself as the clusterhead and broadcast beacons for neighbor detection to obtain connectivity to the sink, so that it can convey its coordinates, time stamp and data plane value to the sink.

IV. SIMULATION AND RESULTS:

We performed the simulations using glomosim [15]. We simulated a hundred homogeneous nodes randomly placed in a 100x100 square meters area. Each node has a transmission range of 55 meters. The initial node placement scenario is depicted in Figure-3.

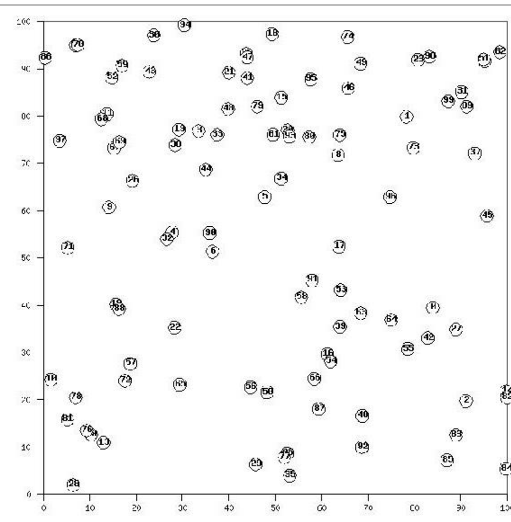


Fig 3. Node placement in Sensor field

A Power:

In the first set of simulations we determine the energy savings obtained from the virtual sense mechanism.

In this set, we simulated 2 scenarios: the single channel and the multi-channel. The nodes are statically divided into 4 different clusters. Each node in this case can transmit directly to the clusterhead. In the single channel scenario every node transmits on the same channel, whereas in the multi-channel scenario a cluster represents a data plane and nodes in a cluster transmit on the same channel but nodes in different planes transmit on different channels and the clusterheads communicate with the sink on the backbone using the control channel 0. Initially all nodes are active. Nodes in the dotted region detect an event and send notification to the sink. In the single channel scenario all the nodes contend for the same channel and send event notification to the clusterheads. The clusterheads in turn also contend for the same channel to relay the notification to the sink. Each clusterhead sends only one notification per event to the sink, that is, the clusterhead discards duplicate notifications, so that the sink receives only one notification per cluster for the same event. In the multi-channel case as a node sends the event notification to its clusterhead, other nodes in the same cluster find the channel busy and refrain from sending any packets for the event. In this case since the event is detected by members of two different planes hence the event is reported twice to the sink, once from each cluster. However, the key point to be noted is that each clusterhead also receives only one notification each, instead of multiple notifications. This saves the transmission energy for all the other nodes in the cluster. It also results in less contention. Since the clusterheads transmit on a separate channel, there is less contention on that channel too.

We varied the sample rate between 1 and 5 seconds with a one second increment and for each case we averaged over multiple values of the seed. For each case we collected the following statistics:

1. The total transmission energy consumed,
2. total number of packets transmitted/received,
3. total number of duplicate packets generated

The results obtained are shown in Figures 4, 5 and 6.

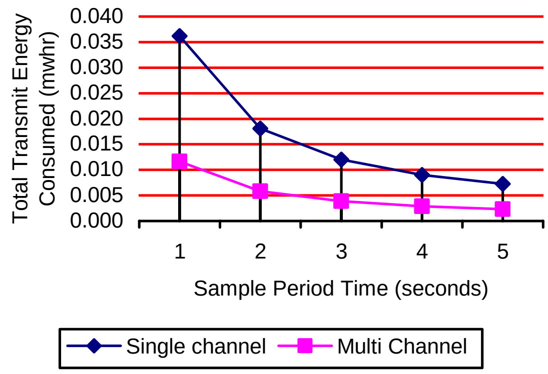


Fig 4: Transmit energy consumed (mwhr) vs. sample rate (seconds)

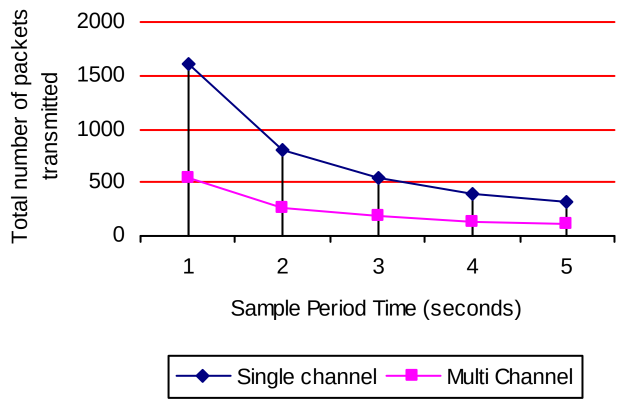


Fig 5: Average number of packets transmitted vs. sample rate

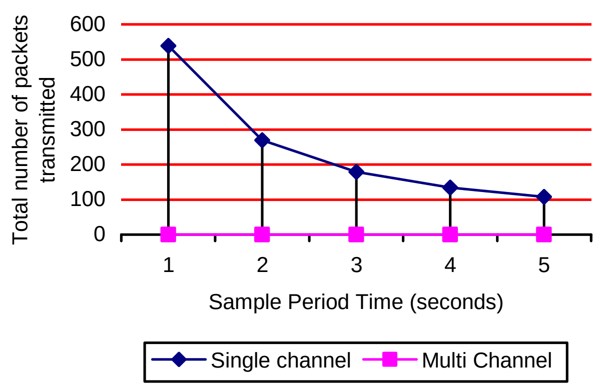


Fig 6: Average number of duplicate packets vs. sample rate (seconds)

These results clearly show that the number of packets transmitted is more than twice in the single channel implementation as compared to the multi-channel implementation for each sample rate. As the number of events per unit time increases, the total number of packets transmitted also increases for each scenario which is to be expected since more number of event notification packets will be generated.

The energy expenditure for transmitted packets in the single channel scenario is also more than three times that of the multi channel scenario. Furthermore, the number of packets generated and the resulting energy expenditure, is directly proportional to the number of nodes that are actively transmitting the event notification packet. The virtual sense mechanism greatly reduces the number of nodes transmitting the event notification packets which in turn reduces the energy expenditure and network traffic in the same proportion. The total energy expenditure may seem to be small. However, that is so because we are simply generating event notification packets which are small. Depending on the data required from the sensor nodes per event, the energy requirement per transmission will increase accordingly and if the virtual sense mechanism is used, it will result in substantial energy savings.

We have considered a relatively simplistic scenario, as the number of data planes increases the number of channels will also increase or/and channels will be reused. Either scheme will incur a higher degree of interference which may result in packet loss lowering the data accuracy. We intend to work on more scenarios to determine the threshold where the energy savings start diminishing.

B. Modeling a Moving Phenomenon:

In the second set of simulations, we modeled a moving phenomenon as a moving curve. As the curve moves, nodes change plane if they have changed position with respect to the line of the curve at each sample period. If a node is, at a sample period, above the line of the curve and then at the next sample time it falls below the curve then it will change its plane and join the other cluster. In this manner nodes can track the movement of the phenomenon. If a node changes plane, it increases its sample rate to a configurable higher value and if after that it remains in the same plane for consecutive configurable number (default is five) sample times it lowers its sample rate. The change in sample rate may be done as a function of the time period that the field is static, however for the purpose of this simulation we only used two values. As nodes join a plane, some of them need to be part of the tree of the new clusterhead in order to maintain connectivity. In our current simulations, we allow new nodes to become tree members based on a probability value. Ideally, only those nodes should become tree members in the new cluster as required to maintain connectivity with all the nodes. We compared the actual scenario at a point of time with the sink's view at that time in order to determine the accuracy of the information being relayed to the user/application. We used the same node placement as shown in Figure-3. However, the clustering is now based on the node positions with respect to the curve. Nodes in a cluster communicate in a channel different from that of nodes in the other cluster.

In Figures 7 to 12, the grey nodes belong to the data plane above the curve Data Plane 1 (DP1) and the white nodes belong the data plane, DP2 below the curve. As the curve moves upwards, nodes from the DP1 join DP2. The figures show the view of the sink at different time instances. At any instance the grey nodes are those that the sink thinks are in DP1 even though according to the curve it should now have changed plane. As we can see from the view of the sink, this set up follows the phenomenon depicted by the moving curve fairly accurately. The error per erring node (that is a node for which there is a mismatch between the actual scenario and the sink's information regarding the plane it belongs to) can be measured as the distance between the ‘y’ coordinate of the node and the ‘y’ coordinate of the curve. The error is directly proportional to the speed of the curve and inversely proportional to the sample rate.

V. CONCLUSION

In this paper we presented a novel distributed clustering mechanism that naturally divides the sensor nodes into cluster based on their data correlation. This is achieved by allowing nodes that sample data which falls in the same data range to communicate on a cluster specific channel. Using this clustering scheme we derived a virtual sense mechanism wherein nodes in the same data plane need not sample data if their channel is busy since some other node within the cluster is already reporting the event. We show that the virtual sense mechanism results in significant energy savings, low network traffic and hence lower contention. We further applied this mechanism to tracking a moving phenomenon (water current, fire) with favorable results.

To the best of our knowledge this is a new approach in sensor clustering field. This approach opens up a multitude of research problems on scheduling and tree pruning to further streamline this mechanism and obtain even more energy savings. The desired accuracy of the sampled data or in other words the error range can be converted into sample rate. This sample rate can be applied to cluster rather than a single node. Within a cluster the sample rate has two dimensions: temporal and spatial. By temporal we mean the rate at which each node samples data, while spatial means the number of nodes that working in tandem can achieve this rate. Given the sampling rate and the number of nodes in a cluster, we can decide how many nodes have to sample for how long to achieve this sample rate in an energy aware manner. Hence given a sampling rate we can translate this into a scheduling problem to achieve a balanced load distribution in the sensor network in order to maintain connectivity and coverage. We are currently researching this problem.

In this paper each cluster corresponds to a data plane. To promote scalability especially in multi-modal sensing environments without sacrificing the granularity we will have to divide the data planes into more than one cluster to enable channel reuse in different clusters. Efficient channel subdivision, assignment and synchronization across clusters is another issue for future research.

Currently we assume that once the channel subdivision is done, we use only so many channels, but we may need further subdivision, based on the data granularity or we may need to reduce the total number of sub-channels for bandwidth optimization for low granularity. Hence dynamic channel subdivision schemes need to be explored more.

As the shape of the data plane changes, the tree structure will have to change too, with the clusterhead moving towards the center and the tree members changing hands with other nodes in order to maintain connectivity. We are looking into efficient tree reformation and pruning algorithms and timely error free handovers

REFERENCES

[1] C. Raghavendra and S. Singh, "Pamas power aware multi-access protocol with signaling for ad-hoc networks," *in ACM Computer Communications 1999, 1999.*

[2] A. P. Samir Goel and T. Imielinski, "Using buddies to live longer in a boring world," *Rutgers Department of Computer Science Technical Report DCS-TR-558, 2004.*

[3] J. W.Ye and D.Estrin, "An energy-efficient mac protocol for wireless sensor networks," *in INFOCOM 2002, 2002.*

[4] T. Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," *in SenSys 2003, 2003.*

[5] B. K. G.Lu and C. Raghavendra, "An adaptive energy efficient and low- latency mac for data gathering in sensor networks," *in WMAN 2004,.*

[6] S. R. Tao Zheng and V. Sarangan, "Pmac: An adaptive energy-efficient mac protocol for wireless sensor networks," *in IEEE WMAN 2005, 2005.*

[7] S. Soro and W. B. Heinzelman, "Prolonging the lifetime of wireless sensor networks via unequal clustering," in *IEEE WMAN 05, 2005*.

[8] R. S. P. Sinha and V. Bharghavan, "Mcedar: Multicast core-extraction distributed ad hoc routing," *in IEEE Wireless Communication and Networking Conference (WCNC), 1999.*

[9] J.Liu, J. and J.Reich, P. Cheung, and F. Zhao, "Distributed group management for track initiation and maintenance in target localization applications," *in International Workshop on Information Processing in Sensor Networks(IPSN), 2003.*

[10] A. C. W. Heinzelman and H. Balakrishnan, "Low energy adaptive clustering hierarchy," *in Proc. Hawaaian Intl Conf. on System Science, January 2000, 2000.*

[11] Curt Schurgers, Vlasois Tsiatsis and Mani B. Srivastava, "STEM: Topology Management for Energy Efficent Sensor Networks," *in IEEE Aerospace Conference'02, Big Sky, MT, March 2002*.

[12] Y. Xu, J.Heidemann, D. Estrin, "Geographically informed energy conservation for ad hoc routing:," *in MobiCom 2001, Rome, Italy, pp. 70-84, July 2001*.

[13] Michael Hempel, Hamid Sharif and Prasad Raviraj, "HEAR-SN: A New Hierarchical Energy-Aware Routing Protocol for Sensor Networks, " *in Proc. Hawaiian Intl Conf. on System Science, 2005.*

[14] D. Petrovic, R. C. Shah, K. Ramchandran and J. Rabaey, "Data Funneling: Routing with Aggregation and Compression for Wireless Sensor Networks," *in IEEE Sensor Network Protocols and Applications (SNPA) 2003, Anchorage, AL, May 2003*

[15] GloMoSim at http://pcl.cs.ucla.edu/projects/glomosim

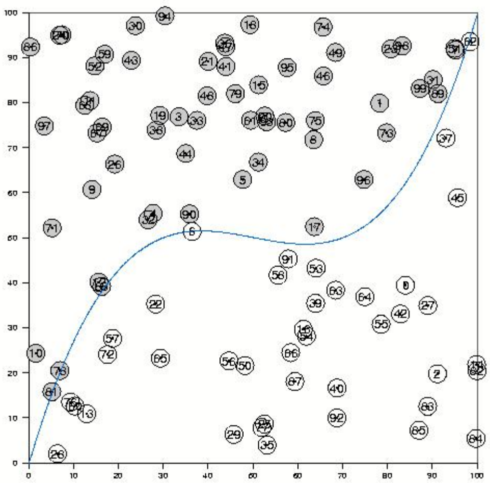


Fig. 7: Time = 0.0

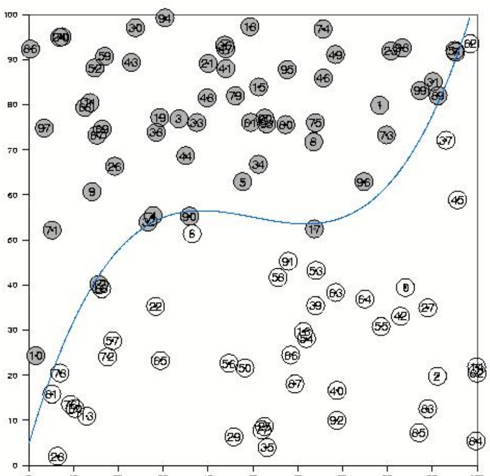


Fig. 8: Time = 10.0

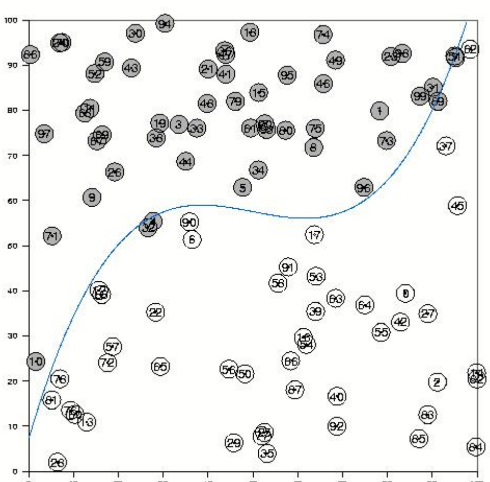


Fig. 9: Time =15.0

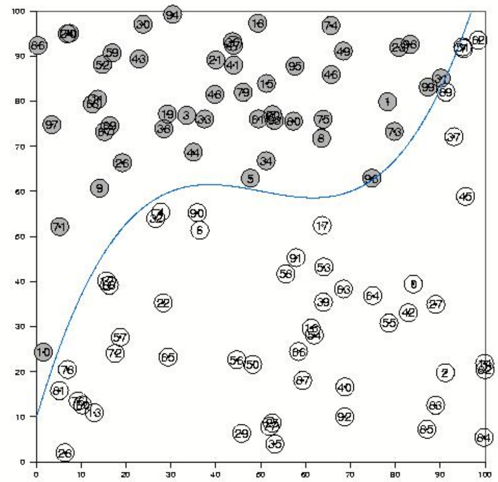


Fig. 10: Time = 20.0

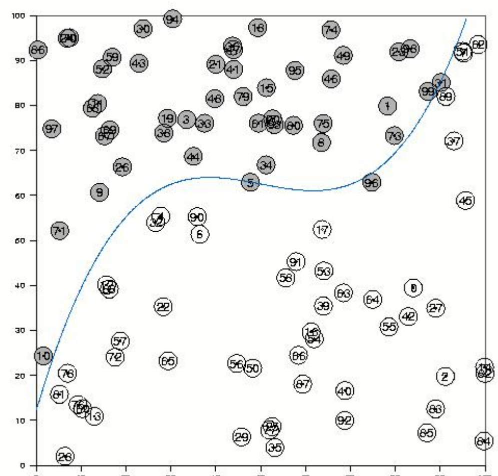


Fig. 11: Time =25.0

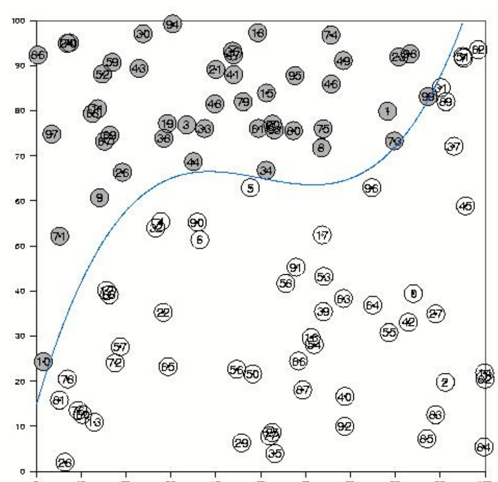


Fig. 12: Time =30.0