NETWORK CLUSTERING AND BOOTSTRAPPING IN WIRELESS SENSOR NETWORKS

[1]L.Shammi, [2]L.Shafnam
[1]Lecturer, Department of Computer Science, KNS Institute of Technology, Bangalore
[2]Assistant Professor, Department of Computer Science, Brindavan College of Engineering, Bangalore

Abstract: Wireless sensor networks have potential to monitor environments for both military and civil applications. Due to inhospitable conditions these sensors are not always deployed uniformly in the area of interest. Since sensors are generally constrained in on-board energy supply, efficient management of the network is crucial to extend the life of the sensors. Sensors‟ energy cannot support long haul communication to reach a remote command site and thus requires many levels of hops or a gateway to forward the data on behalf of the sensor. In this paper we propose an algorithm to network these sensors in to well define clusters with less-energy-constrained gateway nodes acting as cluster heads, and balance load among these gateways. Simulation results show how our approach can balance the load and improve the lifetime of the system.

Keywords: Network clustering, Energy-efficient design, Energy-Aware Communication, Sensor networks.

1. INTRODUCTION

Information gathering is a fast growing and challenging field in today’s world of computing. Sensors provide a cheap and easy solution to these applications especially in the inhospitable and low-maintenance areas where conventional approaches prove to be very costly. Sensors are tiny devices that are capable of gathering physical information like heat, light or motion of an object or environment. Sensors are deployed in an ad-hoc manner in the area of interest to monitor events and gather data about the environment. Networking of these unattended sensors is expected to have significant impact on the efficiency of many military and civil applications, such as combat field surveillance, security and disaster management. Sensors in such systems are typically disposable and expected to last until their energy drains. Therefore, energy is a very scarce resource for such sensor systems and has to be managed wisely in order to extend the life of the sensors for the duration of a particular mission.

Typically sensor networks follow the model of a base station or Command node, where sensors relay streams of data to the command node either periodically or based on events. The command node can be statically located in the vicinity of the sensors or it can be mobile so that it can move around the sensors and collect data.

In either case, the command node cannot be reached efficiently by all the sensors in the system. The nodes that are located far away from the command node will consume more energy to transmit data then other nodes and therefore will die sooner. In order to conserve energy to communicate with the command node various multi-hop and energy aware routing techniques have been suggested in the literature. These techniques have overhead due to route discovery and to find optimum hops to communicate with the command node. In addition, there will be extra burden on the nodes, which are located around the command node as most of the traffic will be routed through them. As a result, these nodes will consume their energy faster than other nodes and will die sooner.

To avoid these overheads and unbalanced consumption of energy we propose an approach where we have some high-energy nodes called “Gateways” deployed in the network. These Gateways, group sensors to form distinct clusters in the system and manage the network in the cluster, perform data fusion to correlate sensor reports and organize sensors by activating a subset relevant to required missions or tasks.

Each sensor only belongs to one cluster and communicates with the command node only through the gateway in the cluster.

Similar to other communication networks, scalability is one of the major design’s quality attributes for sensor networks. A multi-gateway architecture is required to cover a large area of interest without degrading the service of the system. But, if the sensors and gateways are not uniformly distributed then it can cause few gateways to overload with the increase in sensor density, system missions and detected targets/events. Such overload might cause latency in communication and inadequate tracking of targets or events. In order to avoid such patches of dense clusters we balance the load at clustering time to keep the density of the clusters uniform. Simulation results shows that if the load is not balanced among different clusters then few gateways will be heavily loaded after clustering and will consume their energy very soon. This will lead to either all the sensors belonging to that cluster to be inactivated or re-clustering of the system. Frequent reclustering will require extra overhead and change in the system setup. Gateways will have to reschedule the sensors and recalculate the communication paths. Also these additional nodes will add extra burden on all the other gateways.

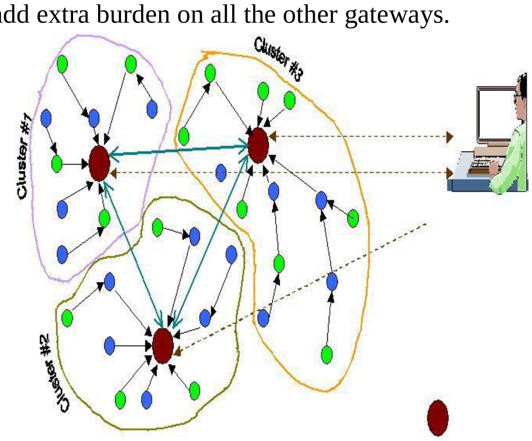

Figure 1: Networking clustering in Wireless Sensor

In the next two sections we define the architectural model of sensor network and summarize the related work. Section 4 describes our approach to load balance clustering of the sensor network. Description of the simulation environment and analysis of the experimental results can be found in section 5. Finally section 6 concludes the paper and discusses our future research plan.

2. SYSTEM MODEL

The system architecture for the sensor network is shown in Fig 1. There are only two kinds of nodes in the system; sensors and less-energy-constrained gateway nodes. The sensors and gateways are assumed to be of the same kind and have same properties respectively. All communication is over wireless links. A wireless link is established between two nodes only if they are in range of each other. Links between two sensors are considered bidirectional while links between a gateway and sensor can be unidirectional depending on the range of the sensor. Gateways are capable of long-haul communication compared to the sensors and all gateways are assumed to be in communication range with one another. Communication between nodes is over a single shared channel. Current implementation supports TDMA protocol to provide MAC layer communication.

In this paper we assume that the sensor and gateway nodes are stationary. In the future we plan to incorporate mobile gateways in the system. During the bootstrapping process, all the sensors and gateways are assigned unique IDs, initial energy and TDMA schedule. The TDMA schedule is only valid for the first phase of the clustering, after that the assigned gateways provide new schedule to the nodes in their clusters. All nodes are assumed to be aware of their position through some GPS system. While the GPS consumes significant energy, it has to be turned on for a very short duration during bootstrapping.

Sensors inform the gateways about their location during the clustering process. It is worth noting that most of these capabilities are available on some of the advanced

Sensors, e.g. the Acoustic Ballistic Module from Sen Tech Inc. [2]

2.1 Sensor's Energy Model

A typical sensor node consists mainly of a sensing circuit for signal conditioning and conversion, digital signal processor, and radio links [3][6]. The following

Summarizes the energy-consumption models for each sensor component.

2.2 Communication Energy Dissipation:.

The key energy parameters for communication in this model are the energy/bit consumed by the transmitter electronics (α t), energy dissipated in the transmit op-amp (α amp), and energy/bit consumed by the receiver electronics (α r) Assuming a 1/d2 path loss, the energy consumed is:

$$Etx = (\alpha\ t + \alpha\ amp\ d2) * r \text{ and } Erx = \alpha\ r * r$$

Where Etx is the energy to send r bits and Erx is the energy consumed to receive r bits. Table 1 summarizes the meaning of each term and its typical value

Table 1: Parameters for the communication energy

Term	Meaning
α t,α	Energy dissipated in transmitter and receiver electronics per bit (Taken to be 50 nJ/bit).
α amp	Energy dissipated in transmitter amplifier (Taken = 100 pJ/bit/m 2).
R	Number of bits in the message.
D	Distance that the message traverses.

2.3 Computation Energy Dissipation:
We assume the leakage current model of [12] [5]. The model depends on the total capacitance switched and the number of cycles the program takes. We used parameter values similar to those in [5].

3. RELATED WORK

Recent advancements in integrated circuits have fostered the emergence of a new generation of tiny, inexpensive low-power sensing devices [8]. Many research groups are exploring various issues like energy aware routing [5], sensor coordination [6], and energy saving through activation of a limited subset of nodes [4].

Most of the published approaches for clustering require a group of sensors to agree among themselves on the election of a cluster-head. The selection of a clusterhead can be randomized [6], based on a pre-assigned cluster ID [5], or according to degree of connectivity. These clustering approaches are effective in the field of ad-hoc networks where the nodes are constantly moving and therefore cannot be in communication range with one cluster-head all the time. Sensor networks display different properties than ad-hoc networks. In unattended sensor networks the location of the sensors is fixed once they are deployed.

In addition, neither of these approaches considers topologies where a larger number of nodes are located in range of each other. If traditional clustering approaches like lowest/highest ID or highest connectivity are applied, the same node will be picked as cluster-head every time, resulting in this sensor to drain its energy very fast. Obviously picking a resource-constrained sensor node for acting as a processing element and a relay for the cluster will quickly deplete its energy. Therefore, some approaches proposed rotating the role of the cluster-head among other sensors in the cluster, based on a round-robin strategy, node connectivity or battery energy. However changing the topology of clusters is undesirable as it imposes huge cluster-head changeover overhead. All other cluster-heads have to be notified about the change. The cluster -heads have to change the scheduling and routing tables to communicate with the nodes.

Moreover these protocols do not consider any balancing of load among clusters. System can have patches of high-density clusters and very low-density clusters. In such scenarios the high-density cluster-head will be overwhelmed with processing and communication load and will consume its energy soon, while the low density cluster-head will sit idle wasting precious time.

4. LOAD-BALANCED CLUSTERING

The main objective of our approach is to cluster sensor network efficiently around few high-energy gateway nodes. Clustering enables network scalability to large number of sensors and extends the life of the network by allowing the sensors to conserve energy through communication with closer nodes and by balancing the load among the gateway nodes. Gateways associate cost to communicate with each sensor in the network. Clusters are formed based on the cost of communication and the load on the gateways.

Network setup is performed in two stages; „Bootstrapping‟ and „Clustering‟. In the bootstrapping phase, gateways discover the nodes that are located within their communication range. Gateways broadcast a message indicating the start of clustering. We assume that receivers of sensors are open throughout the clustering process. Each gateway starts the clustering at a different instance of time in order to avoid collisions. In reply the sensors also broadcast a message with their maximum transmission power indicating their location and energy reserve in this message. Each node discovered in this phase is included in a range set per gateway.

In the clustering phase, gateways calculate the cost of communication with each node in the range set. This information is then exchanged between all the gateways. After receiving the data from all the other gateways each gateway start clustering nodes based on the communication cost and the current load on its cluster. When the clustering is over, all the sensors are informed about the ID of the cluster they belong to.

Since gateways share the common information during clustering, each sensor belongs to only one cluster. For inter-cluster communication all the traffic is routed through the gateways.

4.1 Problem Formulation

Each gateway constructs a range set of all the nodes that can communicate with it. A sensor „Sj‟ belongs to range set „RSet‟ of gateway „Gi‟ if it satisfies the following criteria:

$$Sj \in RSetGi \Leftrightarrow [(RGi > dSj\text{->}Gi) \Lambda (RSj,max > dSj\text{-->}Gi)]$$

Where, R Gi is the range of gateway Gi, RSj,max is the maximum range of sensor Sj and dSj->Gi is the distance between sensor Sj and Gateway Gi. Each node in the range set is associated with a communication cost calculated by the gateway. The cost calculated is a function of communication energy dissipated in transferring „r‟ bits of data over a dSj->Gi distance. Using the energy model described in section 2.1, the cost „Cj,Gi‟ associated with sensor Sj, calculated by gateway Gi is:

$$Cj,Gi = Etx + Erx= (\alpha\ t + \alpha\ amp\ (dSj\text{-->}Gi)2\ ) * r + \alpha\ r * r$$

A cluster record is created with all the nodes in RSet and their associated cost. The record is exchanged by all the gateways to gain global knowledge of the network. Depending on the range of the sensors there can be two kinds of nodes in the system: the first can only communicate with one gateway, „exclusive nodes‟ and the second can communicate with more than one gateway.

The reach of a node is defined as the number of gateways it can communicate with. The first step towards clustering is to separate exclusive nodes from the rest, because these are compulsory nodes to be accommodated by a gateway. To do so, gateways construct another set called exclusive set „ESet‟ which consists of nodes that can satisfy the following criteria:

$$Sj \in ESetGi \Leftrightarrow [(Sj \in RSetGi) \Lambda (\forall\ k\neq i,\ Sj \notin RSetGk)]$$

The load on a gateway is a function of processing load
„PLGi‟ and communication load „CEGi‟ due to sensors in the cluster and is defined as:

$$LGi =f(PLGi , CEGi)$$

Processing load on a gateway is due to processing the data from the sensors in the cluster and energy consumed in doing so. Communication energy, „CEGi‟ of a gateway is calculated to be the summation of the communication cost of all sensors in the cluster. That is,

$$CEGi = \Sigma Cj,i$$

Since, we assume that all sensors are identical and produce data at the same rate, PLGi of a gateway is directly proportional to the number of sensors „n‟ in the cluster. It implies that, to balance load in the system we have to balance the number of nodes in a cluster and the communication energy required per gateway. In order to keep the system close to the average load, we choose an objective function that minimizes the variance of the cardinality of each cluster in the system.

That is, $\sigma 2=1\Sigma G(X - X')Gi\ 0$

Where „σ 2 ‟ is the variance of load in the system, X is cardinality of gateway Gi and X’ is the average cardinality including the node under consideration, G is the total number of gateways in the system.

4.2 Optimization Heuristics

Before minimizing the objective function we allocate the nodes in the ESet to their respective clusters and calculate the load. If we allocate the remaining nodes to the clusters only by minimizing the objective function we experience large overlapping of clusters. Considering only the load on gateways as a factor for clustering might do so at the expense of sensors. Our experiments also show that some sensors are not part of the gateways nearest to them. This will increase the communication energy of the sensors.

Exhaustive search methods like simulated annealing can be used to find the optimum results to balance the Load as well as maintain the minimum distance with the gateway. But by using these methods the complexity of the algorithm is increased with the increase in sensors and gateways. In order to balance load of gateways and preserve precious energy of sensors, we select few nodes that are located radically near a gateway and include them in the ESet of the gateway. A node is included in the ESet of a gateway if, its distance to the gateway is less than a critical distance. Initially, the critical distance is equal to the minimum distance in the ESet. Then the critical distance is gradually increased till the median of distances in ESet is reached. This procedure is repeated for all the gateways based on increasing order of cardinality, which balances load while performing the selection. Experimental results show that this method significantly reduces the number of nodes to be considered for exhaustive search and reduces overlapping between the clusters.

Now, we start clustering the remaining sensors in the system. Since sensors cannot reach all the gateways, minimizing objective function for those gateways will unnecessarily increase the complexity of algorithm. In order to save computation for clustering we sort the sensors based on increasing order their reach. Nodes with same reach are grouped together to avoid extra computation of calculating the objective function for the gateways they cannot reach. Nodes with lower reach are considered first because they have fewer clusters to join. The objective function is calculated by assigning these nodes to the gateways they can reach. The node becomes the part of the cluster for which it minimizes the objective function. The process is repeated till all the nodes in the sensors are not clustered. Fig 2. includes the pseudo code of the algorithm.

```
// This loop creates the ESet of each gateway
 For node = = 1 to N
    If ( node → reach == 1)
        GID = node→ Gateway
        ESet[GID].add (node)
  End If
End For
 // Sort nodes of ESets of all the gateway in ascending order Sort_Set (ESet)

 // Expands ESet based on sesnor‟s distance from Gateway
   For gateway = 1 to G
      Critical_Distance = Min_Distance( ESet[gateway] )
      While (Critical Distance != Median (ESet[gateway])
             node = RSet[gateway] → next
             Distance = Distance(RSet[gateway] → node)
             If( Distance < Critical_Distance)
              ESet[gateway].add(node)
               For GID = 1 to G
                If (RSet[GID].search(node) == TRUE)
                    RSet[GID].remove(node)
                End If
              End For
            End If
        End While
      Critical_Distance = ESet[gateway] → next
End For
// Allocates the nodes of ESets to their respective Clusters or gateway = 1 to G
     While(ESet[gateway] != Enpty)
       node = ESet[gateway] → next
        Load[Gateway] += node Node
    → cluster = gateway
   End While
End For
//This loop creates the groups of same order of reachability
For node = 1 to N
      If ( node → cluster != NULL)
          Group(node→ reach).add(node)
       End If
  End For
// Cluster remaining nodes and balance loads among gateways min_reach = Min_Reach(Group)
max_reach = Max_Reach(Group) For
reach = min_reach to max_reach
     node = Group(reach) → next
    While( node != NULL)
           gateway = Minimize_Objective(node)
           Load[gateway] +=
            node Node → cluster = gateway
    End While
End For
```

Figure 2: Pseudo code for the clustering algorithm

5. CONCLUSIONS AND FUTURE WORK

In this paper we have introduced an approach to cluster unattended wireless sensors about few high-energy gateway nodes and balance load among these clusters. The gateway node acts as a centralized manager to handle the sensors and serves as a hop to relay data from sensors to a distant command node. If nodes are not uniformly distributed around the gateways the clusters formed will be of varied load, which will affect the lifetime and energy consumption of the system. Simulation results demonstrate that our algorithm consistently balances load among different clusters and performs well in all distributions of nodes. Our future plan includes extending the clustering model to allow gateway mobility. Also, we plan to study different failure scenarios in sensor networks and introduce fault-tolerance by providing backup gateways in the system.

REFERENCES

[1] R. Burne, et. al, "A Self-Organizing, Cooperative UGS Network for Target Tracking," Proc. of SPIE Conference on Unattended Ground Sensor Tech. and Applications II, Orlando, April 2000.

[2] "Data sheet for the Acoustic Ballistic Module", SenTech Inc., http://www.sentech-acoustic.com/

[3] M. Bhardwaj, et. al, "Upper Bounds on the Lifetime of Sensor Networks", In Proceedings of ICC 2001, June 2001.

[4] W. Heinzelman, et. al, "Energy-Scalable Algorithms and Protocols for Wireless Microsensor Networks," Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP '00), June 2000.

[5] A. Sinha and A. Chandrakasan, "Energy Aware Software", Proceedings of the 13th International Conference on VLSI Design, pp. 50-55, Calcutta, India. January 2000

[6] B. Chen, et al., "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks", Proc. of MobiCom 2001, Rome, Italy, July 2001.v

[7] Y. Xu and J. Heidemann and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," Proc. of MobiCom 2001, Rome, Italy, July 2001.

[8] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring Sensor Networks Topologies," Proc. INFOCOM 2002, New York, June 2002.

[9] Singh and M.woo and C.S.Raghavendra "Power routing in Mobile Ad Hoc Networks" proc.of ACM MOBICOM‟98,Dallas,Texas,October1998