Attribute Based Hierarchical Clustering in Wireless Sensor Networks*

W. Ke, P. Basu, S. Abu Ayyash, and T.D.C. Little
Department of Electrical and Computer Engineering
Boston University, Boston, Massachusetts
{*ke,pbasu,saayyash,tdcl*}*@bu.edu*

MCL Technical Report No. 03-24-2003

Abstract–Data routing in large wireless sensor networks face the challenge of delivering data over a network in which the nodes may not have globally unique identifiers, must satisfy stringent energy saving requirements and be highly scalable and fault tolerant. In addition, if the sensor network becomes a resource shared by members of a large user community (a likely event in the future), then the routing scheme must also be energy efficient when handling requests that may: (1) arrive at high rates, (2) need different types of data in the response, and (3) need response from subsets of the deployed sensors that satisfy certain attributes.

In this paper we show that a system which uses pure flooding for triggering data collection is energy inefficient under the conditions above. We propose an attribute based hierarchical clustering algorithm as a solution and show that it is more efficient than broadcast related techniques under the circumnstances described. We cluster based on common (i.e. equal) attribute values shared among some sensors, which help contain the broadcast traffic generated to deliver the requests. Furthermore, Our algorithm implements clusterhead failure recovery mechanism and load balancing by rotating the clusterhead functionality among members in the cluster.

Keywords: Clustering, Algorithms, Sensor Networks

*This work was supported by the NSF under grant No. ANI-0073843. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1 Introduction

In these days a typical home or automobile has already $10^1 - 10^2$ embedded computational elements, and in the future the ubiquity of such elements with wireless communication capacity will likely increase. These devices, coupled with sensing hardware, form networks that become the remote "eyes" and "ears" of a large community of users who probe the network looking for data describing the phenomena under observation.

To coordinate data delivery in such large scale networks, in which nodes are assumed not to have globally unique identifiers, data centric models, such as Directed Diffusion [1] are often recommended. In such models, requests for data for the phenomena observed are flooded from one node, the base station, to the entire network. The flooding process establishes unicast routes from the sensors to the base station, and sensors that have data of the phenomena requested respond. Note that because routing is "data-centric," all the local "routes" are tracked with respect to specific data requested. This means that if multiple requests arrive, each querying about a different type of data or about sensors that satisfy a specific attribute value, multiple floodings are necessary to establish the reverse unicast routes to the base-station, even if the sensors that can answer all queries are few in number and are located within the same small geographic region.

We present a theoretical model we use in our energy expenditure analysis in Sec. 2. We compare the cost effectiveness of query delivery through flooding schemes and through schemes that actively maintain a structure. We show the conditions under which systems that use flooding are inefficient, and we show that such conditions map to a sensor network which is heavily utilized and which is shared among a community that does not favor querying all the nodes in the sensor network. We delineate our hierarchical clustering algorithm in Sec. 3 and conclude in Sec. 4.

2 Theoretical Model

From the user's perspective, a sensor network is the medium through which requests for data regarding a phenomenon can be made and responses expected [2]. Thus a fundamental primitive that represents users accessing the sensor network is a query, that is, a specific request for the values of certain type of data (temperature, humidity, etc.) from sensors that satisfy certain attributes (temperature sensors, sensors in a certain geographic location, etc.). In our paper we use the term "inquiry" as a more encompassing term that includes queries and other primitives for accessing the sensor network. Thus a community of users can be represented as inquiries that arrive with a certain rate λ, in which the requests Q obey a probability distribution function $P(Q)$ of requesting data within the space in which the sensor network is deployed. The cost in our model is the number of transmissions required to deliver the inquiry.

We make the following simplifications before proceeding to some theoretical analysis: (1) we assume that answers to inquiries traverse through unicast routes in both schemes (flooding and the hierarchical clustering) and their costs are the same and (2) the variable λ refers to the rate of arrival of requests for data of a type not queried previously and/or from sensors of a different attribute, i.e., requests that trigger a flooding in the flooding-based schemes.

In these schemes, a wireless network composed of N sensors, deployed over total time T, incurs the following expected cost C_{flood} for inquiry delivery: $C_{flood} = \lambda_D T N$.

On the other hand, a scheme that actively maintains a structure L (L represents a structure which has a measurable maintenance cost) incurs different cost values when answering different inquiries. Such scheme's expected cost $C_{structure}$ during the deployment time T is:

$$C_{structure} = C_{maintenance}(N, L, T) + \lambda_D T \int_{\mathbf{Q}} C(L, N, Q) P(Q)\, dQ$$

$C_{maintenance}$ is the cost of maintaining the structure L for N sensors during T. From the two equations above, we see that $C_{structure}$ will be less than C_{flood} when: (A) $\int_{\mathbf{Q}} C(L, N, Q)\, P(Q)\, dQ < pN$, $0 \leq p < 1$, and (B) $C_{maintenance}(N, L, T) < (1 - p)\lambda_D T N$. (A) is true when the cost of delivering one inquiry in the presence of the structure L is less than that of one flooding (N). That is, when the arriving inquiries Q do not need to reach all the nodes in the sensor network (otherwise flooding is necessary). This is likely to happen when multiple groups share the sensor network with different research objectives. The maintenance cost $C_{maintenance}$ represents the cost needed to maintain the structure L during T and thus is invariant with respect to the number of inquiries that arrive. (B), therefore, is satisfied when λ is high enough, that is, when there is a large enough community of users actively using the sensor network, resulting in the high arrival rates of inquiries for different data.

Thus when a sensor network is shared by a large community of users of diverse objectives, a structured format for inquiry delivery which incurs a fixed maintenance cost can be more energy efficient than purely flooding mechanisms. In view of the analysis above, we propose clustering the sensors according to attributes that are meaningful to the inquiries, e.g., location, and have the clusterheads relay the inquiries[1]. If we establish a hierarchy of such attributes, in which clusters at higher levels contain lower level clusters, we gain control over which sensors receive inquiries. Energy and bandwidth are saved when top level clusterheads drop irrelevant inquiries while relevant inquiries are forwarded to the appropriate clusterheads and flooded inside the cluster. The process of designing and specifying the hierarchies is beyond the scope of this paper. In the next section we describe briefly our clustering algorithm. We assume the existence of an attribute hierarchy, which we interchangeably call "containment hierarchy" (CH).

3 Attribute Based Clustering

The algorithms we developed form same-attribute clusters with one clusterhead and rotate the clusterhead functionality among cluster members. Cluster sizes are constrained whenever possible, so as to avoid managing disproportionately large clusters. Devices with higher energy levels are selected in the clusterhead rotation process. One algorithm can be seen at Alg. 1. Other algorithms have been omitted due to space limitations.

Cluster Formation We propose a modified *leader* algorithm [3] to form clusters. Alg. 1 describe the specific mechanisms of our clustering algorithm when a sensor receives a packet indicating the start of the clustering process. If the sensor determines it will be a clusterhead

[1]Ideally the structure L minimizes C_S, but such optimal structure may be hard to find.

(cluster leader) candidate for any hierarchy level, it will store this information, activate a timer (time-out value inversely proportional to energy level), and send out its candidacy packet upon timer expiration. All cluster formation decisions are localized decisions and all clusters across all hierarchy levels are formed in one network-wide flooding. This flooding is part of the maintenance cost which is independent of the inquiry arrival rate, and which is "amortized" as this clustered structure is re-used to deliver new incoming inquiries.

Algorithm 1 Cluster Formation Algorithm

```
1: Initialize Processed, Stored, Timer, Candidacy;
2: On receive packet P, P.type = CLUST_FORM
3: if (P ∉ Processed) then
4:   for (∀ CH levels L) do
5:     if (My.L.leader = ∅) then
6:       if (My.L.attribute = P.L.attribute) then
7:         if (P.hop ≥ max∧ (no lower CH level ∨ lower CH level changes attribute)) then
8:           add L to Candidacy; Stored ← P; start Timer ∝ 1/My.Energy;
9:         else
10:          accept leadership information from P.L;
11:      else
12:        add L to Candidacy; Stored ← P; start Timer ∝ 1/My.Energy;
13:    else
14:      if (P.L.leader more suitable) then
15:        accept leadership information from P.L;
16:      else
17:        delete P.L in P;
18:  if (∀ CH levels L, My.L.leader ≠ ∅) then
19:    cancel Timer;
20:  if (Stored = ∅) ∧ (∃ P.L ≠ ∅) then
21:    add P.hop by 1; rebroadcast P;
```

Cluster Leader Rotation Leader rotation avoids single devices from being completely energy depleted in their burden in the clusterhead role. The rotation period is adjusted according to the frequency of inquiries arriving at the cluster and to the leader's level in the hierarchy level (higher level leaders rotate less). Rotation takes place when a time-out of the cluster member with the highest energy level takes place (a node's time-out value is inversely proportional to its energy level). This member floods the cluster and becomes the new cluster head in a way similar to the cluster formation process.

Cluster Recovery and Update Algorithms Clusterheads send periodic `ALIVE` messages to its k-hop neighbors (k being a tunable parameter of the algorithm). These neighbors also keep a copy of whatever information the clusterhead is maintaining. The neighbor which detects cluster failure floods the cluster identifying itself as "interim clusterhead" and a rotation mechanism follows. Cluster member failures do not trigger any recovery mechanisms, for we assume the sensor network to be dense enough, in which individual sensor failures do not impair cluster related functions and properties. If that is not true, then peer monitoring among clusterheads may be necessary to recover from partitions in the attribute value region. Newly deployed sensors will attempt to join the "best" neighboring clusters

that have the same attribute values. If no clustered sensors are detected, the new sensors will remain isolated until a cluster formation packet arrives. This mechanism effectively supports dynamic hierarchy level updates, i.e., if there is an addition of a CH level, then sensors receiving the update are effectively "new without-leader" (in that level) sensors which are in an already deployed network. The removal of a level is done by erasing the membership information regarding that level from the cluster members while keeping all the other levels' information intact.

4 Conclusion

In this extended abstract we performed a simple cost effectiveness analysis of flooding based mechanisms for triggering data collection versus mechanisms which actively maintain a structure to deliver inquiries. We showed that under heavy utilization and high degree of sharing among a large community, sensor networks employing pure flooding systems are less efficient. We propose an attribute based hierarchical clustering mechanism as a solution and delineate the properties of our clustering algorithm. Our algorithm is fast, requiring only one network wide flooding to establish all clusters across all hierarchy levels. In addition, it is robust with respect to clusterhead failure, and implements load balancing by rotating the clusterhead functionality among cluster members. Also, the "update" feature allows for changes in the specification of the containment hierarchy, which can be used for dynamic containment hierarchy level optimization.

References

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. "Next Century Challenges: Scalable Coordination in Sensor Networks". In *Proceedings of the 5th ACM MobiCom Conference*, Seattle, WA, August 1999.

[2] P. Bonnet, J. E. Gehrke, and P. Seshadri. "Querying the Physical World". *IEEE Personal Communications*, 7(5):10–15, October 2000. Special Issue on Smart Spaces and Environments.

[3] J. A. Hartigan. *Clustering Algorithms*. Wiley, New York, NY, 1975.