ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

An Effective Way of Cluster Formation Based On Load Rebalancing

S.Padmini,
M.E, Department of CSE., GKM College of Engineering and Technology, Anna University, Tamilnadu ,India

ABSTRACT—A fully distributed load balancing algorithm is presented to cope with the load imbalance problem. Our algorithm is compared against a centralized approach in a production system and a competing distributed solution presented in the literature. The simulation results indicate that our proposal is comparable with the existing centralized approach and considerably outperforms the prior distributed algorithm in terms of load imbalance factor, movement cost, and algorithmic overhead. The performance of our proposal implemented in the Ha-doop distributed file system is further investigated in a cluster environment.

I. INTRODUCTION

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

Due to energy constraints, a sensor node can but communicate directly with other sensors within a limited distance. In order to enable communication between sensors out of each other's communication range, sensors form a multi-hop communication network. Utilizing clustering algorithm to form a hierarchical network topology is the accustomed realizing scenario of network management and data aggregation for WSNs, and clustering facilitates the distributed control of the network. Moreover, clustering nodes into groups not only saves energy but also reduces network contention when nodes communicate their data over shorter distances to their respective cluster-heads. Motivated by the former research of clustering algorithm, the objective of this paper is to study a Balanced Clustering Algorithm with Distributed Self-Organization for Wireless Sensor Networks (DSBCA), which can deal with stochastic distribution of sensor nodes.

II. RELATED WORK

Leach protocol Wireless distributed micro sensor systems will enable the reliable monitoring of a variety of environments for both civil and military applications. We look at communication protocols, which can have significant impact on the overall energy dissipation of these networks. Based on our findings that the conventional protocols of direct transmission, minimum-transmission-energy, multi-hop routing, and static clustering may not be optimal for sensor networks, we propose LEACH (Low-Energy Adaptive Clustering Hierarchy), a clustering-based protocol that utilizes randomized rotation of local cluster base stations(cluster-heads) to evenly distribute the energy load among the sensors in the network. LEACH[1] uses localized coordination to enable scalability and robustness for dynamic networks, and incorporates data fusion into the routing protocol to reduce the amount of information that must be transmitted to the base station.

HEED Approach is used to Topology control in a sensor network balances load on sensor nodes, and increases network scalability and lifetime. Clustering sensor nodes is an effective topology control approach. A novel distributed clustering approach for long-lived ad-hoc sensor networks. The HEED(Hybrid Energy-Efficient Distributed

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

clustering), that periodically selects cluster heads according to a hybrid of the node residual energy and a secondary parameter, Such as node proximity to its neighbors or node degree. HEED[8] terminates incurs low message overhead, and achieves fairly uniform cluster head distribution across the network. We prove that, with appropriate bounds on node density and intra-cluster and inter-cluster transmission range, HEED[8] can asymptotically almost surely guarantee connectivity of clustered networks.

Geographic Adaptive Fidelity or GAF [10] is an energy aware location-based routing algorithm designed primarily for mobile ad hoc networks, but is used in sensor networks as well. This protocol aims at optimizing the performance of wireless sensor networks by identifying equivalent nodes with respect to forwarding packets. In GAF protocol, each node uses location information based on GPS to associate itself with a "virtual grid" so that the entire area is divided into several square grids, and the node with the highest residual energy within each grid becomes the master of the grid. Two nodes are considered to be equivalent when they maintain the same set of neighbor nodes and so they can belong to the same communication routes. Source and destination in the application are excluded from this characterization.

Nodes use their GPS-indicated location to associate itself with a point in the virtual grid. Inside each zone, nodes collaborate with each other to play different roles. For example, nodes will elect one sensor node to stay awake for a certain period of time and then they go to sleep. This node is responsible for monitoring and reporting data to the sink on behalf of the nodes in the zone and is known as the master node. Other nodes in the same grid can be regarded as redundant with respect to forwarding packets, and thus they can be safely put to sleep without sacrificing the "routing fidelity".

WCA: A Weight Clustering Algorithm

The network formed by the nodes and the links can be represented by an undirected graph $G = (V,E)$, where V represents the set of nodes vi and E represents the set of links ei. That the cardinality of V remains the same but the cardinality of E always changes with the creation and deletion of links. Clustering can be thought as a graph partitioning problem with some added constraints. As the underlying graph does not show any regular structure, partitioning the graph optimally (i.e., with minimum number of partitions) with respect to certain parameters becomes an NP-hard problem [7].

More formally, we look for the set of vertices $S \subseteq V(G)$, such that $\_v \in S\ N[v] = V(G)$.The neighborhood of a cluster head is the set of nodes which lie within its transmission range. The set S is called a *dominating set* such that every vertex of G belongs to S or has a neighbor in S.

We show that the overlapping multi-hop clustering problem is NP-hard and propose the **k**-hop **O**verlapping **C**lustering **A**lgorithm (*KOCA*) as a randomize distributed algorithm for solving it. *KOCA[15]* aims at generating connected overlapping clusters that cover the entire sensor network with a desired average number of boundary nodes in the overlapping area. *KOCA* operates in stationary networks where nodes do not move and have equal significance, which is a reasonable assumption for sensor networks.

Nodes randomly elect themselves as cluster heads with some probability p. The cluster head probability (p) is a given parameter to the algorithm that can be tuned to control the number of overlapping clusters in the network. After the termination of 2 the clustering process, each node is either a cluster head or within k-hop from *at least one* cluster head, where k (*clusterradius*) is another given parameter to the algorithm. The clustering process terminates in O(1) iterations, independent of the network size, and does not depend on the network topology or size. Through analysis and simulation experiments we show how to select the different values of the parameters to achieve the clustering process objectives. Moreover, the results show that the *KOCA* algorithm incurs low overhead in terms of messages exchanged and produces approximately equal-sized clusters, which allows distributing the load evenly over the different cluster heads.

III . SYSTEM MODEL

Load balancing algorithms help you easily fine-tune how traffic is distributed across connections. Each deployment has a unique setup, and Peplink's enterprise grade load balancing features can fulfil all of your special requirements. Create your own rule with the following algorithms and you can sit back and enjoy the high performance routing that Peplink brings to you.

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

LOAD REBALANCING ALGORITHM

In our proposed algorithm, each chunk server node I first estimate whether it is under loaded (light) or overloaded (heavy) without global knowledge. A node is *light* if the number of chunks it hosts is smaller than the threshold.

Load statuses of a sample of randomly selected nodes. Specifically, each node contacts a number of randomly selected nodes in the system and builds a vector denoted by V. A vector consists of entries, and each entry contains the ID, network address and load status of a randomly selected node.

Algorithm:

Load balancing algorithms help you easily fine-tune how traffic is distributed across connections. Each deployment has a unique setup, and Peplink's enterprise grade load balancing features can fulfil all of your special requirements. Create your own rule with the following algorithms and you can sit back and enjoy the high performance routing that Peplink brings to you.

Weighted Balance:

Assign more traffic to a faster link or less traffic to a connection with a bandwidth cap.

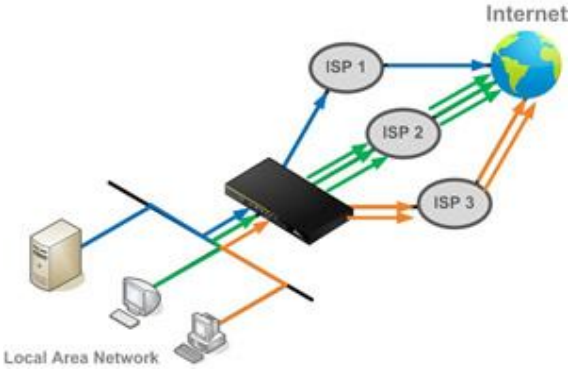


Fig: 4.1.1 WEIGHTED BALANCED

Network Load Balancing facilitates the process of creating a Web Server Farm. A Web Server farm is a redundant cluster of several Web servers serving a single IP address. The most common scenario is that each of the servers is identically configured running the Web server and whatever local Web applications running on the Web server as shown in Figure 1. Each machine has its own copy of everything it needs to run the Web application which includes the HTML files, any script pages (ASP, ASP.Net), any binary files (such as compiled .Net assemblies, COM objects or DLLs loaded from the Web app) and any support files such as configuration and local data files (if any). In short the application should be fully self-contained on a single machine, except for the data which is shared in a central location. Data typically resides in a SQL backend of some sort somewhere on the network, but could also be files shared in a directory for files from a file based database engine such as Visual FoxPro or Access.

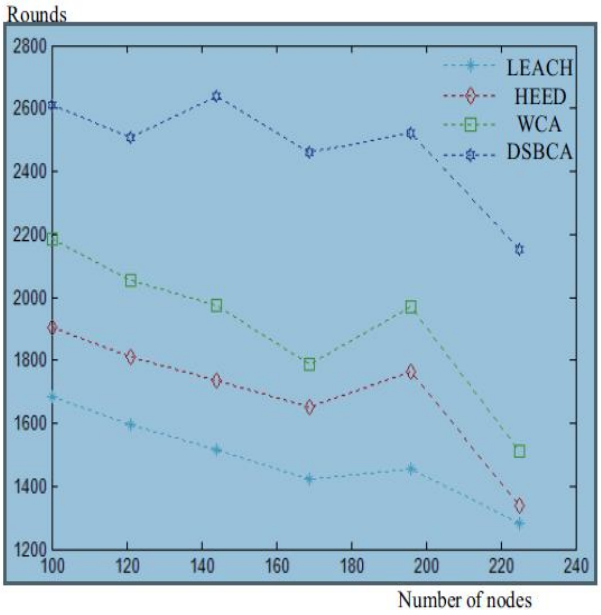
Load statuses of a sample of randomly selected nodes. Specifically, each node contacts a number of randomly selected nodes in the system and builds a vector denoted by V. A vector consists of entries, and each entry contains the ID, network address and load status of a randomly selected node.DSBCA sets the threshold of cluster size. The number of cluster nodes cannot exceed the threshold to avoid forming large clusters, which will cause extra overhead and thus reduce network lifetime. When the cluster head node receives Join message sent by the ordinary node, it will compare the size of cluster with threshold to accept new member and update the count of cluster nodes if the size is smaller than threshold, or reject the request.

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014



VI. PROBLEM OVERVIEW

Many distributed real-time applications, such as audio- and video-conferencing, require the network to construct a multicast path (tree) from a sender to multiple receivers. Furthermore, real-time applications have Quality-of-Service (QoS) requirements (e.g. bandwidth). The objective of the routing protocol is to build a tree that is both feasible (i.e. satisfies the requested QoS) and least costly. The cost of a tree depends on the costs of its links. The cost of a link should reflect the effect of allocating resources to the new connection on existing and future connections. Many studies have proposed multicast algorithms to construct low-cost QoS-constrained trees. However, these studies assume that some cost is given for each link, and they do not examine the effect of the link cost function.We examine the effect of various link cost functions on the performance of two classes of multicast routing algorithms under both uniform and skewed real-time workload. We also investigate the impact of inaccurate network state information. we find that, a link cost function that is more sensitive to load performs better when link state information is relatively accurate, while when link state information is more stale, a link cost function that is less sensitive to load performs better. This is more pronounced under higher load and when multicast groups are denser.

V . PROPOSED WORK

DSBCA clustering in uniform distribution.

The purpose of DSBCA is to generate clusters with more balancedenergy and avoid creating excessive clusters with many nodes. The clusters near the base station also forward the data from further clusters (all clusters need to communicate with the base station, but long-distance wireless communication consumes more energy), and as we all know, too many members in a cluster may bring about excessive energy consumption in management and communication. Hence, based on the above concerns, DSBCA algorithm considers the connectivity density and the location of the node, trying to build a more balanced clustering structure. The basic idea of DSBCA is based on the connectivity density and the distance from the base station to calculate *k*(clustering radius). The clustering radius is determined by density and distance: if two clusters have the same connectivity
density, the cluster much farther from the base station has larger cluster radius; if two clusters have the same distance from the base station, the cluster with the higher density has smaller cluster radius. DSBCA forms different clustering layers in which the radiuses of farther clustering layers are larger, and in the same layer the clustering radius identical

DSBCA clustering in non-uniform distribution.

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

In the case of non-uniform distribution, the cluster radiuses are determined by the distance from the base station and connectivity density of nodes. With farther distance from the base station and lower connectivity density, the cluster radius is larger; on the contrary, with closer distance from the base station and lower connectivity density, the cluster radius is smaller.

VI. IMPLEMENTATION

DSBCA follows a distributed approach to establish hierarchical structure in self-organizing mode without centralcontrol. DSBCA selects the random nodes to trigger clustering process first. Then the trigger node *Ut*calculates it's connected

To density and distance from the base station to determine cluster radius *k* and becomes the temporary cluster head.

$k = floor[\beta D(Ut)/Dk(Ut)]$

where $D(u)$ is the distance from the base station of u, $Dk(u)$ is the connectivity density of node u, $\beta$is the sensor parameters determined by specific applications of WSNs, and *floor* is the

calculation of rounding.DSBCA follows a distributed approach to build hierarchicalstructure in self-organizing mode without central control.

The cluster is stable for a while until the process ofreselecting cluster head is triggered in $T(k)$. The cluster head gathers the weight of all member nodes, and then selects the node with highest weight as the next head node. In this way, the communication costs are decreased. The reelecting of cluster head occurs in the 'old' cluster, so the broadcast of temporary head and the corresponding responses of all the *k*-hops neighbors are unnecessary.

VII . CONCLUSION

The curve of DSBCA is smoother than others, which indicatesthat the clustering structure is more stable. So DSBCA could adapt to to diverse distribution. However, the life cycle of other traditional algorithms drops sharply when the number of nodes exceeds 150. The decline in the is relative to the clustering structure and the selection of cluster head. DSBCA can form more reasonable cluster structure to avoid frequent exchange of the nodes weight information and temporary cluster head broadcasting after the first clustering.

As a result, the energy consumption decreases effectively. The cluster structure changes in each round in LEACH, HEED and WCA; nevertheless, DSBCA maintains relatively stable clustering structure in which switching of cluster head often occurs in the same cluster.

ACKNOWLEDGEMENT

We wish to express our sincere thanks to all the staff member of Computer Science and Engineering Department, GKM College of Engineering and Technology for their help and cooperation.

REFERENCES

[1] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "Energyefficientcommunication protocol for wireless micro sensor networks,"
inProc. 33rd Annu. Hawaii Int. Conf. Syst. Sci., Jan. 2000, pp. 1–10.

[2] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," IEEE Trans. Mobile
Comput., vol. 3, no. 4, pp. 366–379, Dec. 2004.

[3] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in Proc. 7th Annu. Int. Conf. Mobile
Comput.Netw., Jul. 2001, pp. 70–84.

[4] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A weighted clustering algorithms for mobile ad hoc networks," Cluster Comput., vol. 5, no. 2,pp. 193–204, 2002.

[5] Y. Fernandess and D. Malkhi, "K-clustering in wireless ad-hoc networks," in Proc. 2nd ACM Workshop Principles Mobile Comput. Conf., Oct. 2002, pp. 31–37.

[6] H. C. Lin and Y. H. Chu, "A clustering technique for large multi-hop mobile wireless networks," in Proc. 51st Int. Conf. Veh. Technol., May 2000, pp. 1545–1549.

[7] M. Lehsaini, H. Guyennet, and M. Feham, "A novel cluster-based self-organization algorithm for wireless sensor networks," in Proc. Int. Symp.
Collabor.Technol. Syst. Conf., May 2008, pp.1926.

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

[8] M. Youssef, A. Youssef, and M. Younis, "Overlapping multihop clustering for wireless sensor networks," IEEE Trans. Parallel Distrib.Syst., vol. 20, no. 12, pp. 1844–1856, Dec. 2009.
[9] N. Mitton, B. Sericola, S. Tixeuil, E. Fleury, and I. G. Lassous, "Selfstabilization in self-organized wireless multihop networks, Ad Hoc Sensors Wireless Netw., vol. 11, no. 2, pp. 1–34, 2011.
[10] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," Mobile Comput.Commun. Rev.,