American Journal of Information Science and Computer Engineering
Vol. 1, No. 3, 2015, pp. 107-112
http://www.aiscience.org/journal/ajisce

A Power Efficient Cluster Head Election Algorithm for Mobile Ad-Hoc Networks

Dominic Damoah, Edward Danso Ansong*, Asamoah Kwaku Acheampong

Faculty of Science, Department of Computer Science & Information Technology, Valley View University, Accra, Ghana, West Africa

Abstract

Mobile ad-hoc networks (MANETs) provide a medium of routing from a source to a destination through the use of multi-hop routing. Clustering mechanisms are used to organize the network topology in a hierarchical manner. Networks of this nature usually adapt the use of radio frequencies to transmit packets. Moreover, some nodes within this setup usually relies on battery power, therefore, rendering the whole setup power dented. This paper proposes an algorithm that utilizes important parameters such as number of nodes, power consumption per packet transmission and proximity to determine the cluster head selection for efficiently energy liveliness. The algorithm was researched, designed, coded, implementation and tested in a simulated environment for thoroughly checks. The tool used for the simulation is Net Logo 5.1.0 which is a domain expert platform for the programming non expert. The algorithm enhances the operations of cluster computing in ad-hoc environments efficiently with regards to cluster head selection.

Keywords

Cluster Head, MANETs, Routing, Wireless Transmission, Ad-Hoc Network, Simulation

Received: May 21, 2015 / Accepted: June 8, 2015 / Published online: July 13, 2015

1. Introduction

There have been numerous proposals on how routing should be handled in such a setup. The metrics used to actually handle clustering and transmission of data diversifies the various algorithms. Each algorithm is designed to perform its task as well as possible with using a single or multiple metric.

The cluster head serves as the coordinator in a clustering setup. It is tasked with a myriad of operations. Therefore, it is imperative to select a node that could handle these tasks efficiently in the setup in order to keep the cluster stable and efficient as possible. The aim of this paper is to propose an algorithm that aims at utilizing important parameters specifically, number of nodes, power consumption per packet transmission and proximity.

In the last decade, we have been witnesses to the rapid change in computer technology. Computers nowadays can be said to be pervasive in our daily lives. In the network arena, wireless connectivity is becoming a common technology as many user-agents employ its usage. The existence of a wireless network structure without the presence of a base station saw the coinage of a more prudent system called "MANET" (Mobile Ad-Hoc Network). MANETs allow high performance of protocols for the MAC layer by improving throughput, scalability and power consumption [1]. Clustering algorithms are therefore implemented to create and maintain cluster [2]. A cluster is typically made up of nodes but these nodes differ. They are *ordinary nodes*, *cluster heads* and *gateway nodes*. Ordinary nodes do not have any affiliation to nodes in different clusters. Gateway nodes have no cluster head and are located at the boundary of a cluster. They are tasked with inter cluster communication. In the process of packet transmission from source to destination, there is always the problem of getting the shortest possible path. Therefore, an efficient system should consider the

* Corresponding author
E-mail address: kwddamoah@gmail.com (D. Damoah), edkan20002002@yahoo.com (E. D. Ansong), asamoahak@live.co.uk (A. K. Acheampong)

shortest path for a packet to be transmitted [3].

In MANETs, routing is a tricky concept as compared to the traditional wired networks. The above statement holds true due to the fact that the network topology changes at a fast pace. Therefore, the use of conventional routing protocols may not satisfy this setup due to their large computational time and its non-compliance with ever changing topology [4]. In order to curb the rate of packet loss due to ever-changing topology, routes in MANETs need constant calculation [5]. The life time of a MANET is usually determined by the total amount of time the first node runs out of energy [6].

2. Related Work

A number of protocols take their foundation from an already existing protocol known as the LEACH (Low Energy Adaptive Clustering Hierarchy) in their quest to mend the cluster head election process [7]. Some limitations of LEACH identified indicate that scalability is low when network increases in size, possible cluster head concentration is always in a single area, energy consumption is assumed to be equal among nodes, and wild assumption that all node have a direct connection to the base station.

Another algorithm to be considered is the Lowest ID Cluster Algorithm (LIC). The nodes in this setup are assigned unique identifiers (IDs). A node becomes the appropriate cluster head only if its ID is the lowest among other nodes [8]. The IDs are illogically assigned numbers. A node broadcast their IDs and the node that has the least ID becomes the cluster head. The major flaw of this scheme is that a node serves as a cluster head for a longer period of time. Moreover, no other parameter is used in determining the cluster head hence may render a cluster less efficient in terms of power consumption.

The Highest Degree Algorithm scheme uses the distance between nodes to assign degrees or IDs to a node. If the Euclidean distance between the two nodes is less than the range, then there is a link between those nodes. The node with the maximum degree is chosen as the cluster head among other nodes [9]. Due to the absence of a threshold value for total number of nodes within the cluster, some clusters end up being bloated hence the throughput drops which results in a poor overall performance.

The K-hop Connectivity ID Clustering Algorithm (KCONID) can be said to be a hybrid algorithm that uses techniques from both the lowest-ID and highest-degree algorithms [10]. First, the connectivity criteria are used to elect a cluster head. If there exists a clash in connectivity, then the second criteria, lowest-ID is invoked. These schemes aim at creating the minimum number of clusters as possible and also obtain the smallest dominating set.

On the other hand the Weighted Cluster Algorithm *(WCA)* scheme uses a myriad of parameters to select a node as the cluster head. It can, therefore, be said to be a combined metric scheme. The factors considered includes transmission power, mobility, number of nodes a cluster head can handle etc. [11]. The demerit on this scheme is that the quest for reducing the load on the cluster head instigated the employment of a threshold value to limit the number of nodes each cluster head can handle.

3. Proposed Scheme

The proposed scheme (Bobalgo), utilizes three simple yet important techniques to select a cluster head. The proposed scheme's power biased nature, and employs relevant tested methods used will not be any different. Before the process of selecting a cluster head is done, every node will be ranked using the following metric;

i. Highest number of neighbors

ii. Highest level of residual energy

iii. Multiple cluster association

In a MANET setup, a node has a good global view of the network when it has lots of other nodes as neighbors. This helps the node to intelligently route the packets it receives with ease and also helps reduce the total overhead in the network. Due to this, the node with the highest number of neighbors will store its total number of neighbors in a memory (table). A diagram of nodes and neighbors is depicted in figure 1.

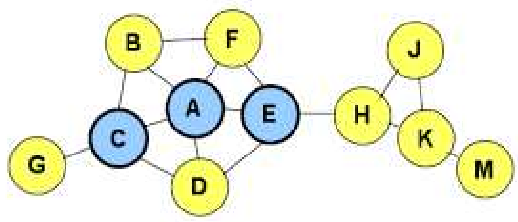

Figure 1. A diagram of nodes and neighbours.

Next to be considered is the energy level remaining in the nodes power pack. Residual energy is used because it tends to be a good metric. First off, there is no efficient way to check how long a node will survive in the setup. Nodes use energy to carry out activities like routing and sometimes even movement. Therefore, each node will keep the rate at which it is losing energy every five (5) minutes. This will be calculated as the (current total energy - residual energy after 5 minutes / 2). The rate will also be placed in memory. When a node finds itself wandering between two clusters, it becomes eligible to be the agent to route packets among the two clusters. This also promotes the idea that the node in question has a greater reach than other nodes. Greater reach is a good characteristic for cluster head selection. This means packets can be routed to such a node from the farthest edges of the network. A diagram of a node that falls between two clusters is shown in figure 2.

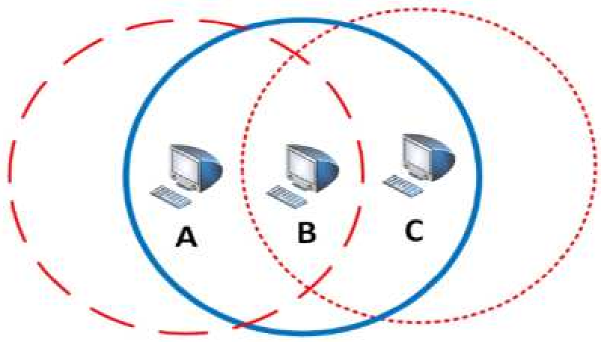

Figure 2. A diagram of a node that has multi-cluster associativity.

Therefore, any node which can be associated with the two or more clusters also stores the number of clusters it can associate itself with in memory.

4. The Election

The number of nodes within a MANET setup may vary from time to time. Due to this, the proposed algorithm will only initiate elections between two nodes at a time. When the election process is triggered, two nodes closest to each other weigh their parameters. The node that comes up on top broadcasts its ID to its neighbors. Each and every neighbor stores the IDs and marks of all the nodes that came up first in these two candidate node election. Afterwards, the cluster nodes initiate the election process by comparing the marks of the winner nodes. The winner nodes will not be part of this process. The one that tops the process will have its ID stored and exchanged by all the individual nodes aside the winner nodes. When a consensus is reached, the ID of the node that won the process will be propagated throughout the cluster. Each node will acknowledge the ID of the node with the highest rank and stores it in memory as the cluster head. On the other hand when a tie is reached, thus, two cluster nodes having the same rank, a special packet is sent by the two nodes in question. This special packet, "*a throughput packet*" is sent by both nodes. This packet will be broadcasted throughout the cluster. This packet contains data about the time the packet left the winner nodes. Each node will acknowledge this packet upon receiving. The time when each node receives the packet is stored. The average of the time it took for the other nodes to receive the packet will be calculated by the winner nodes. The node with the least average will then assumes the position of the cluster head.

5. Semantics of the Cluster Head Selection

In the simulation, 30 nodes are created at different coordinates in the environment. After this, the each node has an energy level randomly generated. Each node then tries to discover its neighbors with respect to its wireless transmission range and creates a connection to the node closest to itself. Nodes are paired virtually due to their one-to-one connection with the node closest. The first phase of the election is triggered. Thus, the weighing of node metrics as defined earlier specifically residual energy, number of neighbors and multi-cluster associativity. When the election process is complete, the cluster head is selected.

6. Detailed Implementation of the Cluster Head Selection Algorithm

The number of cluster nodes in this simulation was limited 30. This was created by the code snippet in figure 3.

The initial setup of the simulation environment was created by a function called *setup()*. The function first of all clears the environment of all objects. Next, it creates 30 nodes and places them randomly in the environment. Upon completion of the above steps, a reset of the timer, *ticks,* is done.

The next part of the simulation process is to give each and every node some level of energy. The energy levels were randomly assigned. The function to do this is called *p_lvl()*. A code snippet of the function is shown figure 4.

The wireless transmission range is set by another function called *trans_range()*. This is to define the range within which a node can talk to other nodes within the specified range. A snippet is shown figure 5.

When the above function is triggered, each node creates a blue splotch around itself. Any node that falls within another node's splotch is a neighbor. The nodes now initiate connection with the node closest to it. This is to pave way for the election process. A connect function named *conn()* shown in figure 6 helps the nodes to accomplish this goal.

The next stage involves the one-to-one election process. The nodes initiate a superiority check at this stage. The metrics used (power, number of neighbours and multi-cluster association) is done here. The process is summarized in Figure 7.

```
to setup ;;setting up environment
  clear-all ;;clearing everything
  create-turtles 30 ;; creating number of nodes
  ask turtles [setxy random-xcor random-ycor] ;; setting nodes at random places
  reset-ticks ;;resetting timer
end
```

Figure 3. A code snippet of simulation environment setup.

```
to p_lvl
  ask turtles[
set energy energy + random 100  ;;initial battery power generated randomly

  ]
end
```

Figure 4. A code snippet for generating node energy.

```
to trans_range ;;setting tranmission range of nodes

  ask turtles[
 ask patches in-radius 4[set pcolor blue]] ;;each node makes a blue splotch around itself

end
```

Figure 5. A code snippet for node transmission range

```
;;creating connection link between nodes
to conn
  let closest-turtle min-one-of other turtles  [distance myself]
  create-link-with closest-turtle

end
```

Figure 6. A code snippet to simulate inter node connection.

7. Results of Simulation

The result is echoed throughout the cluster and the eligible node is chosen as the cluster head. A snapshot of the simulation environment is shown figure 8 when the setup function is initiated.

After the setup, the one-to-one connection among closest nodes is triggered. This is also shown in figure 9.

Figure 10 is another snapshot of nodes displaying their various scores after the election.

After the nodes are done with the voting process, the cluster head is chosen by selecting the node with the highest score as shown in figure 11. This is also depicted below.

```
Initialize each node
        Energy = random number depending on residual
energy
        isHead = false
        hNeigh = empty  //has neighbor
        nScore = 0 //nodes weight or score
        Find neighbor // Nodes within range
                If Neighbor > 1
                nScore+number of neighbours
                Else
                        Act as head
        Broadcast cluster ID
If Node hears > 1 id
        Nscore + number of ids heard
Else
        Nscore --
Select cluster head node (node with highest score)
If tie
        isHead=false
        Send throughput packet
        Calculate Average time (Sent and Received)
If Average time is less
        Assume CH position
```

Figure 7. A code snippet of the cluster head election process.

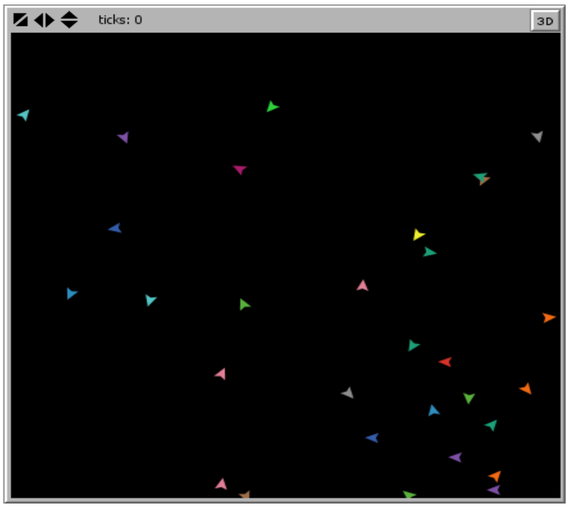

Figure 8. A graphical depiction upon the activation of the setup function.

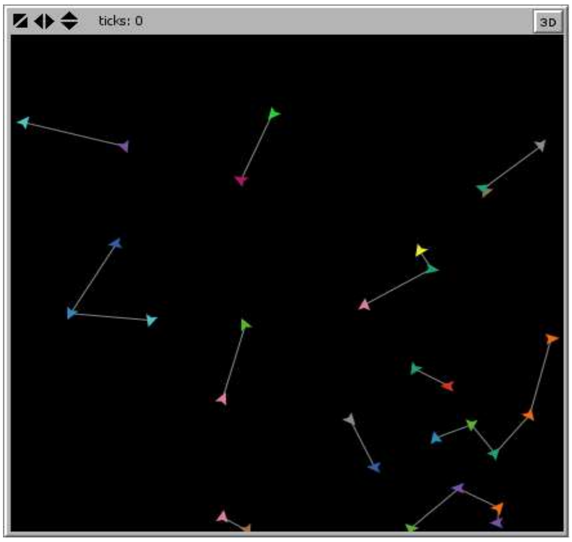

Figure 9. Nodes showing one-to-one associativity.

It can be seen from the snapshot in figure 11 that a cluster head has been chosen by a cluster at the lower left corner of the screen. The cluster head makes a blue splotch around itself.

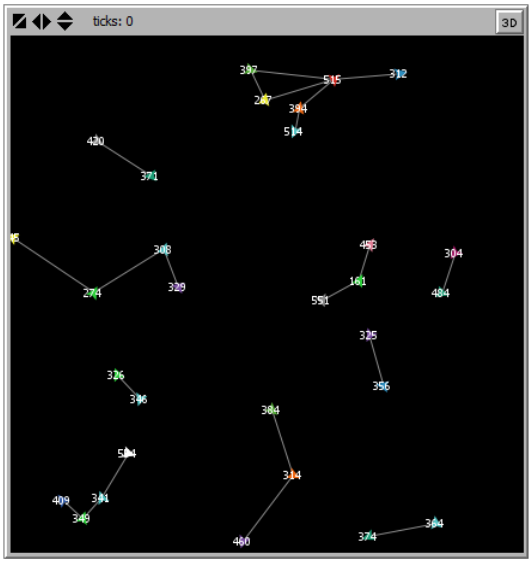

Figure 10. Nodes showing their individual scores.

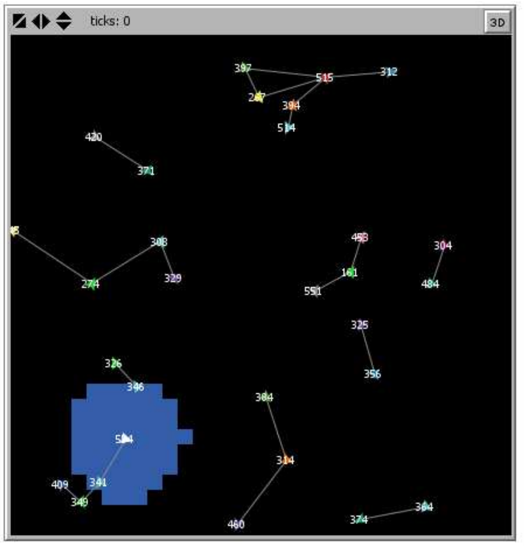

Figure 11. An eligible cluster head selected.

8. Conclusion

Mobile ad hoc network (MANET) has many emerging applications, which include commercial and industrial, war front applications, search and rescue operations, sensor networks and vehicular communications [12]. Although there are a lot of proposed methods to handle this situation are available, it is believed that the most integral part of the system is its cluster head selection methods. If this small sector can be made as efficient as possible, then the whole working performance of the network will be greatly improved.

The presence of transient network links, mobility and limited battery power of mobile nodes in MANETs poses a hefty challenge for such networks to scale and perform efficiently when subjected to varying network condition. [13]

In contribution to the ongoing research about the most effective way to establish a clustering environment with regards to cluster head election in mobile ad-hoc networks. We have investigated wireless technologies in cluster computing, created a new algorithm to efficiently manage battery life in wireless nodes, revamped existing ways of selecting cluster servers (cluster heads) and efficiently managed associating and de-associating nodes over the wireless range.

When fully implemented Network-aware applications obtain information about their execution environment and dynamically adapt to enhance their performance. Adaptation is especially important for synchronous parallel applications since a single busy communication link can become the bottleneck and degrade overall performance dramatically.

9. Future Work

The next step in this research is to add the impact of mobility of the node to the election process. Nodes that are unstable to any cluster may lose some point in the election process. This will be done with a much powerful tool and analysis of how fluid packets travel through the setup will also be considered in the network setup.

References

[1] T. C. Hou and T. J. Tsai, "An access-based clustering protocol for multihop wireless ad hoc networks," IEEE JSAC, vol. 19, no. 7, pp. 1201–1210, 2001.

[2] G. Lugano. (2003) Clustering for hierarchical routing. [Online].

[3] Y. Fernandess and D. Malkhi, "K-clustering in wireless ad hoc networks," in POMC "02: Proc. of the second ACM international workshop on Principles of mobile computing. New York, NY, USA: ACM Press, 2002, pp. 31–37.

[4] P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan, "A clusterbased approach for routing in dynamic networks," SIGCOMM Comput. Commun. Rev., vol. 27, no. 2, pp. 49–64, 1997.

[5] Y. P. Chen and A. L. Liestman, "A zonal algorithm for clustering ad hoc networks." [Online]. Available: citeseer.ist.psu.edu/chen03zonal.html.

[6] Ye Tian, Min Sheng, Jiandong Li, Yan Zhang, Junliang Yao and Di Tang," Energy aware Dynamic Topology Control Algorithm for wireless Ad Hoc Networks",2008 IEEE

[7] M. J. Handy, M. Haase and D. Timmermann, Low energy adaptive clustering hierarchy with deterministic cluster-head selection, in Proc. 4th IEEE International Workshop on Mobile and Wireless Communications Network (MWCN '02), Stockholm, Sweden, September 2002, pp. 368-372.

[8] M. Gerla and J. T.-C. Tsai, "Multicluster, mobile, multimedia radio network," Wirel. Netw., vol. 1, no. 3, pp. 255–265, 1995.

[9] R. Agarwal, M. Matwani, "Survey of Clustering Algorithms for MANET", International Journal on Computer Science and Engineering, Vol. 1(2), 2009, pp. 98-104.

[10] G. Chen, F. Nocetti, J. Gonzalez, and I. Stojmenovic, "Connectivity based k-hop clustering in wireless networks," Proc. of the 35th Annual Hawaii International Conference on System Sciences (HICSS"02)-Volume 7. Washington, DC, USA: IEEE Computer Society, 2002, p. 188.3.

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

[12] Ratish Agarwal et al, "Review of weighted clustering algorithms for mobile ad hoc networks," Computer Science and Telecommunications 2012,no. 1 (33) pp71).

[13] Abhishek Majumder, Nityananda Sarma "DEMAC: A Cluster-Based Topology Control for Ad Hoc networks" International Journal of Computer Science Issues, vol.7, no.5, pp.82-88, 2010.