

# Mobile Ad hoc Networks and its Clustering Scheme

Neva Agarwala<sup>1</sup>

<sup>1</sup> Southeast University, Dept. of EEE,  
Dhaka, Bangladesh  
mnagarwala@seu.ac.bd

**Abstract:** *Adaptability is a key issue in the successful operation of a MANET while Mobile Ad-Hoc Networks (MANET) has many characteristics that are a challenge to manage, like bandwidth and power constraints, dynamic topology, etc. Unfortunately, this feature of MANET have not been investigated and optimized in great detail in the past. Hence, this provides an ideal opportunity and so forms the heart of this thesis. This thesis attempts to initiate such an adaptability study, by introducing a performance metric called Performance Factor (PF), which directly investigates the performance of a link by considering the bandwidth available to a link and distance between the Cluster head and the user node. An algorithm has been proposed that utilizes this performance metric to ensure that all nodes receive optimum performance while ensuring an optimum number of clusters is maintained. All in all, the thesis puts forward a new area of research on MANET, along with a scheme for MANET network to better adapt to the changes in its topology and an analysis that validates the use of this scheme.*

**Keywords:** MANET, Clustering, Multi-hop, Routing Protocols

## 1. Introduction

Since the advent of computers, researchers as well entrepreneurs have invented numerous ways to integrate it into our everyday lives. Consequently, human life has drastically improved since then and does not resemble at all the life-style of the last century.

Once such breakthrough is the quick establishment of communication networks, which have revolutionized how people communicate. Today, no place is too far, and thus the world converges to the possibility immediate communication as when and where it is desired.

While wired networks provide communication services like the Internet over fast transmission mediums like optical fibre, wireless communication is gaining importance of equal value very rapidly. Hence, it is time to realize to the possibilities for the next generation of wireless communication, as this arena of communication is receiving much attention from academia, industry and the government.

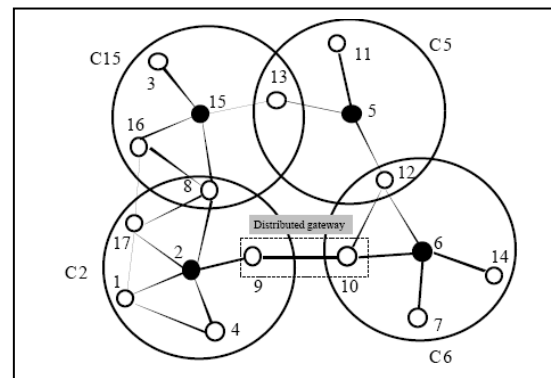
We all know the impact of mobile phones and how it has become an integral part of our lives. Unlike the centrally controlled service of mobile phones, a communication network can be set up 'on the fly'. Such networks are known as Ad Hoc Networks and are the focus of this thesis.

Ad-hoc networks are vastly becoming a lucrative research as well deployment issue since it can be setup as soon as it is needed. This is especially useful when the need of fast deployment of mobile users arises. Consequently, Mobile Ad-Hoc Networks (MANET) brings about numerous applications, such emergency/rescue operations, disaster relief efforts, and military networks and all networks that do not rely on a centralized and organized connectivity [1,2].

## 2. Clustering Architecture

Now, we proceed to describe cluster architecture. Figure 1

[3] illustrates a clustering architecture with labeled nodes and clusters. Two nodes are said to have a link between them if they within transmission range of each other. Each node has a unique identifier, which is its identity. Several designations need to be known here.



**Figure 1.** Clustering Architecture [3]

**Cluster head:** cluster heads forms the identifier of each cluster. Cluster heads regulates channel assignment, power control, bandwidth utilization and time division synchronization. With such significant responsibilities, cluster heads must be chosen carefully; a resourceful algorithm is usually devised. In Figure 1, the nodes in black are cluster heads.

**Gateway nodes:** nodes 13, 12, and 8 in Figure 13 are called gateway nodes. They are an essential part of the cluster architecture since their presence makes inter-cluster possible. As can be seen, gateway nodes are nodes that are within transmission range of two cluster heads. This is a special case of clustering architecture, which contains overlapping clusters.

**Distributed gateways:** to provide for inter-cluster routing in non-overlapping clusters, distributed nodes are used. These nodes are identified since they are members of different clusters that are within transmission range of each other. Nodes 9 and 10 are the distributed gateways of the cluster

architecture in Figure 1.

Ordinary nodes: nodes that are not cluster heads or gateways are referred to as ordinary nodes. Although an ID separates nodes, groups of ordinary nodes belong to a cluster, which forms identification for ordinary nodes. For ordinary nodes 3,8,13 and 16, node 15 is the cluster head and they all belong to cluster C15. [3]

With clustering providing some significant benefits, a new problem arises when we try to come up with simple but proficient algorithms to divide nodes into clusters. There are numerous ways of accomplishing this, but the different factors to consider can be overwhelming. Stability, load balancing, mobility, maximum cluster size, minimum number of clusters, variations in clusters, maximum number of hops to the cluster head, power control, bandwidth utilization and many more aspects need to be optimized. Hence in the pool of clustering algorithms, each considers one or more of these aspects. All these algorithms focus on different problems and so they are suited to different environments and used for different applications.

### 3. Clustering Algorithms

With clustering providing some significant benefits, a new problem arises when we try to come up with simple but proficient algorithms to divide nodes into clusters. There are numerous ways of accomplishing this, but the different factors to consider can be overwhelming. Stability, load balancing, mobility, maximum cluster size, minimum number of clusters, variations in clusters, maximum number of hops to the cluster head, power control, bandwidth utilization and many more aspects need to be optimized. Hence in the pool of clustering algorithms, each considers one or more of these aspects. All these algorithms focus on different problems and so they are suited to different environments and used for different applications.

The following section briefly explains some of the earliest clustering schemes and also ones that are considered relevant.

#### 3.1 Linked cluster algorithm (LCA)

This is one of the simplest algorithms; it dispenses a unique ID to each node. The node with the highest ID is assigned to be the Clusterhead. Hence, nodes that have the highest ID among its neighbors, or are in a neighborhood where it has the highest ID among its neighbors become the cluster head.

An unnecessary number of nodes were elected as a result and this had to be modified into an algorithm called LCA2. The concept of uncovered and covered nodes was introduced where a node belonging to a cluster head was considered covered. Now, a node that has the lowest ID among its uncovered neighbors is elected as the cluster head. [4]

#### 3.2 The Highest Connectivity Clustering Algorithm

This is used in conjunction with the lowest ID algorithm, almost always ensures that the number of clusters formed is the minimum. Nodes here broadcast the number of neighbors they have. This is called the degree of the node. The node with the highest degree, consequently the greatest number of neighbors is elected as the cluster head. In case of a tie, the node with the lowest ID prevails.

Intended for small sized networks with about a hundred nodes at most, this algorithm provides many useful characteristics. First, a large number of cluster heads are not elected. Secondly, topology becomes such that no two cluster heads are directly linked, they are at least two hops away. Finally, all nodes in a cluster are linked with the cluster head.

In many cases, highest connectivity results in some nodes being too frequently elected as cluster heads. Rapid power depletion of the elected node can force it into becoming inactive. To alleviate this problem, a virtual ID (VID) was introduced along with the existing physical ID. The VID is initially set to zero and keeps count of the number of times that node has become cluster head. This reduces the probability that a node with a higher value of VID will become the cluster head.

#### 3.3 CLUSTERPOW algorithm

There are many other algorithms that concentrate on different issues of MANET. Power is of great importance since battery power of communicating node keeps a MANET alive. As a result, power aware clustering have been developed, reviewed in [5] and upgraded with DSDV algorithm in [6]. Each node in a MANET executing CLUSTERPOW algorithm, keeps a routing table that contains information about the transmit power level to other nodes. Power control is used to increase network capacity, decrease the contention of the link layer and save energy. The clustering scheme groups nodes with lowest transmit power levels together. It does not elect cluster heads or make use of gateways, which is its inherent drawback. In addition, overhead required to maintain state information of power levels is too demanding. Chatterjee, Potluri and Negi [5] also sketchily reviews mobility based and weighted based clustering, another two popular clustering schemes. Mobility based clustering of which MOBIC is a prime example examines the mobility behavior of nodes and uses this as the dominant metric for designing the cluster scheme. Each node transmits their aggregate mobility which is the average mobility of all its neighboring nodes. The node with the least aggregate mobility is elected as the cluster head [5][6]. Besides requiring high communication overhead, high latency during cluster formation are disadvantageous.

#### 3.4 Weighted clustering

On the other hand combines a number of metrics like battery life, robustness, bandwidth and SNR to present an efficient algorithm for cluster formation. Ghosh, Das, Som,

Bhattacharya, Venkateswaran and Sanyal [7], proposes such an algorithm and establishes a parameter called the Optimum Node Performance Factor (ONPF) formed from measurements of the battery life, signal strength, bandwidth, signal-to-noise ratio (SNR) and processing speed of nodes. With the ONPF, the algorithm first forms a preliminary set of selected nodes of which, the node with the highest degree becomes the cluster head. The algorithm certainly provides for better Quality of Service (QoS), but the only drawback seems to be the complex computation at each node and the large communication overhead involved.

### 3.5 Domination Set (DS) Clustering

Clustering can also be done by constructing Domination Set (DS), hence, the name DS Based Clustering. [5] briefly examines and [Chen] explores this clustering schemes. The dominating nodes in a DS act as cluster heads and is responsible for relaying routing information and packets. Each node is assigned to a cluster head, which dominates it. A variation of DS is Connected Dominating Set (CDS) in which all of the DS are connected to each other. While this scheme may offer simplicity in routing and maintenance, heavy burden is put on the cluster head node as the workload rapidly eats away its power. This is because inter-cluster routing and forwarding is the sole responsibility of the cluster head, which can only renounce its role only when it has been depleted of its power.

### 3.6 Bandwidth Adaptive Clustering

Bandwidth is another scarce resource that must be carefully managed in MANET. This was the focus of [8]. Here, bandwidth utilization is reduced and managed effectively by determining a forwarding probability based on the available bandwidth. Hence, the more bandwidth is available the more probable that the maintenance messages will be forwarded. BAC calculates a forwarding probability PF determined from the available bandwidth percentage  $P_{avl}$ . Since the amount of overhead determines the bandwidth to be employed, BAC also tackles this problem by reducing the amount of message overhead. Yet another way to save bandwidth resources is to ensure that a MANET is divided into the least number of clusters possible, thus introducing the concept of minimizing the number of clusters. This can be accomplished by using a predefined cluster size  $U$  having both upper and lower bounds to balance workload for cluster heads and the Join and Merge operation as proposed in [8]. As nodes in a cluster leave or enter the system, there are situations where the number of nodes may be too small and a separate cluster structure for these few nodes results in inefficiency. In such cases, ordinary nodes as well as cluster head nodes may require joining other clusters. Ordinary nodes use the Join operation to connect to a cluster head of another cluster, whereas a cluster head decides on behalf of all the other nodes in its cluster and selects an adjacent cluster to join. The cluster head then requires a Merge operation to request to join the adjoining cluster and if successful notifies its member nodes to join the same cluster. This way, too small clusters are avoided by combining

clusters in whole and the number of clusters is minimized.

### 3.7 Minimizing Number of Nodes

Lastly, an algorithm, which concentrates solely on minimizing the number of clusters have been proposed by Sheu and Wang in [3]. The algorithm first places a restriction on the degree of the cluster head, keeping nodes with degree  $n$  less than or equal to  $D_{avoid}$  ( $n \leq D_{avoid}$ ), from becoming cluster head. By using a weighted value obtained from  $D_{avoid}/n$ , the non-clustered degree ( $nc\_degree$ ) and the ID of each node the cluster head is determined. Each node computes all three of these parameters by sending HELLO messages to their neighbors. The node with the highest weighted\_value becomes the cluster head; in case of a tie, the node with the highest  $nc\_degree$  assumes the position; if a tie occurs again, the node with the highest degree prevails.

## 4. The Proposed Algorithm

### 4.1 Assumptions

The model assumed before designing the proposed algorithm has the following assumptions:

- The network is assumed to be static and the average relative mobility is assumed to be 0. Hence mobility will not play a role in the algorithm and so a mobility model is not adopted.
- New clusters need not be formed; the algorithm does not employ cluster splitting; only cluster merging is dealt with.
- At several stages a predefined number of nodes is assumed to be in the network. Here the simulation usually begins with 20 nodes in the setup phase.
- Area of Investigation equal to 10mX10m.
- All nodes that are active has packets to send.
- Distances of nodes are used as a replacement for transmission range. In real-life scenarios each node knows its transmission range and the use of GPRS is not needed.

### 4.2 Performance Factor (PF)

A metric has been devised based on which the algorithm is designed. The PF has been designed to judge the QoS of a link. It comprises of the bandwidth available for the link and the distance between the node and its Clusterhead. Hence, the Performance Factor is a quantitative value that provides the quality of a link. Evidently, the higher the value of PF means the better the service.

From experience, it is understood that the higher the bandwidth the better the QoS. On the other hand, the lower the distance between the node and the Cluster head, the better the QoS. Therefore, the Performance Factor can deduced to be directly proportional to the bandwidth and inversely proportional to the distance. Hence,

$$\text{Performance Factor(PF)} = \text{Baverage/Distance}$$

where,

$B_{average} = B_{available} / \text{number of nodes}$

The parameter  $B_{average}$  is required because the bandwidth of a cluster is assumed to be assigned to the Clusterhead, since it is the Clusterhead that routes a packet from source to destination and so it is the Clusterhead that allocates the resources to each of its member nodes.

### 4.3 Description of the Proposed Algorithm

The proposed algorithm has been devised in the following three stages:

- The Setup Phase deals with the formation of clusters.
- Node Birth describes how the scheme adapts when new nodes joins the network.
- Node Death describes how the scheme adapts when nodes leave the network.

#### 4.3.1 The Setup Phase

The algorithm for the Setup Phase minimizes the number of clusters by electing those nodes having an optimum number of neighbors; this number is termed as  $D$ . The Setup Phase proceeds as follows:

- All nodes transmit a HELLO packet with a Time to Live (TTL) value of 1. This HELLO packet will expire after traversing a distance of one-hop count, thus traveling to its immediate neighbors.
- Upon receipt of these HELLO messages, nodes will reply with a HELLO\_ACK packet.
- By calculating the number of these packets received, a node can find the number of neighbors it has.
- Nodes that determine its number of neighbors to be equal to  $D$ , will broadcast a CLUSTERHEAD( $C\_ID$ ) packet. The Clusterhead ID is the same as the ID of the node.
- Nodes receiving this packet can proceed to determine which Clusterhead to join with. They do so by calculating its distance from each of the Clusterhead.
- The node joins with the Clusterhead that is closest to the node by sending a JOIN( $N\_ID$ ) packet, registering with the Clusterhead, its ID.
- The join is confirmed when the Clusterhead replies with a JOIN\_ACK( $C\_ID$ ,  $N\_ID$ ) packet. The  $C\_ID$ ,  $N\_ID$  values are sent for verification purposes.
- If the Clusterhead decides not to accept the node, it sends a JOIN\_REJECT( $C\_ID$ ,  $N\_ID$ ) packet. In such cases, the node will further attempt to join with the

Clusterhead that is the next minimum distance away from it.

- After this instigation, all Clusterheads determines the Performance Factor (PF) of each of its neighbors as outlined in Section 12.2.
- The Clusterheads finally transmits the PF to its respective member nodes by sending a PERFORMANCE( $C\_ID$ ,  $N\_ID$ , PF). The nodes can now compare its received performance to its desired performance, if this facility is available at the user interface.

Thus, at the end of the Setup Phase, the appropriate Clusterhead have been decided, all nodes know which Clusterhead and ultimately which cluster it belong to, along with the level of performance it is going to receive.

#### 4.3.2 Node Birth

After the Setup Phase, the notion of adaptability can now be modeled. As stated before, in this thesis, adaptability is investigated in terms of node birth and death in the system. The proposed algorithm adapts to node birth in the following ways:

- The new nodes first send a broadcast packet SEARCH( $N\_ID$ ) packet through the entire network. Only the Clusterheads of the network replies to the SEARCH packet by sending a SEARCH\_ACK( $C\_ID$ , PF). The PF in the SEARCH\_ACK packet is calculated by each Clusterhead as follows:

$B_{average} = B_{available} / \text{number of nodes already in the cluster}$

$\text{Performance Factor (PF)} = B_{average} / \text{distance between the new node and the Clusterhead}$

- Each of the new node attempts to join with the Clusterhead that offers the maximum PF by sending a JOIN ( $N\_ID$ ) packet.
- The Clusterhead in turn replies with a JOIN\_ACK( $C\_ID$ ,  $N\_ID$ ) packet for confirmation or JOIN\_REJECT( $C\_ID$ ,  $N\_ID$ ), to reject the join operation.
- In case of rejection, the node attempts to join with the Clusterhead that offers the next highest PF.

It is evident that with new nodes joining a cluster, the performance to previously existing nodes will be reduced, as the number of nodes increases. Hence, each Clusterhead should now recalculate the PF to all its member nodes according to the steps defined in Section 2.2.

Hence, it is seen that the add operation taking into account the bandwidth available to each Clusterhead as well as the

distance, defines how new nodes should combine with the network.

### 4.3.3 Node Death

With the delete operation, the algorithm adapts to network dissolution. It specifically attends to two scenarios, by restricting the minimum cluster size, defined by MIN\_SIZE. The deletion operation works as follows:

- All nodes belonging to a cluster periodically transmit a HELLO message to its Clusterhead. This signifies that the member is existent in the cluster and receives service from its Clusterhead.
- If a HELLO message is not received after the specific period of time expires, the Clusterhead assumes that the node has left the cluster and a node death has occurred.

The Clusterhead then determines whether the existence of the cluster is justifiable. It follows the following procedure:

- If the number of nodes in the cluster is greater than MIN\_SIZE, the Clusterhead simply recalculates the Performance Factor (PF) for the nodes that are left, since the number of nodes in the cluster has reduced.
- If the number of nodes in the cluster is less than or equal to MIN\_SIZE, the existence of this cluster is not justified and the following steps are followed. The algorithm corresponds to the one used in Section 12.3.2 and is outlined again for convenience:
- Each of the remaining nodes send a broadcast packet SEARCH(N\_ID) packet through the entire network.
- The other Clusterheads of the network replies sends a SEARCH\_ACK(C\_ID, PF). The PF in the SEARCH\_ACK packet is calculated by each Clusterhead as follows:

$B_{average} = B_{available} / \text{number of nodes already in the cluster}$

$PF = B_{average} / \text{distance between the new node and the Clusterhead}$

- Each of the remaining nodes attempts to join with the Clusterhead that offers the maximum PF by sending a JOIN (N\_ID) packet.
- The Clusterhead in turn replies with a JOIN\_ACK(C\_ID, N\_ID) packet for confirmation or JOIN\_REJECT(C\_ID, N\_ID), to reject the join operation.

- In case of rejection, the node attempts to join with the Clusterhead that offers the next highest PF.

Finally, the Clusterhead with the new members recalculates the PF for each of its member nodes since the number of nodes in the cluster has increased. The delete phase employs a minimum bound size to prevent too small clusters from occurring; otherwise bandwidth as well as other resources may be utilized inefficiently.

## 5. Software Used

For implementing the algorithm described in Section 12.3, Microsoft Visual C++ was used. This proved to be a good choice, as the language is simple. Since this algorithm calls for significant amount of tracking, this was made easy by employing structures and functions. Altogether, the resulting code can be understood by the most novice of programmers.

Furthermore, Microsoft Excel was used to store and keep track of the test data, from which the graphs in Section 15 were plotted. The test data used is provided in Appendix B.

To produce the graphs, a graphing utility was used. Graph, Version 4.1 provided a smooth way to plot a series graph from the test data. Section 15 comprises of these graphs and their analysis.

## 6. Simulation Parameters

For implementation and obtaining the test data, several Simulation Parameters were used.

First of all, the initial number of nodes had to be specified and this was set at 20 nodes. Later as the investigation continued, the number of nodes varied and the maximum number of nodes increased up to 50 nodes. Conversely, when the test data for the deletion operation needed to be obtained, nodes were deleted in steps to a minimum of 35 nodes.

In the program as well as the algorithm, an one\_hop count was used to determine the number of neighbors in each node. This hop\_count is equivalent to the transmission range of the nodes whose average value is determined to be 20m, in an area of 10mX10m.

The Performance Factor (PF) is comprised of the bandwidth available to a Cluster head. This bandwidth is assumed to be approximately 70Mbps to each Cluster head, with around 10Mbps allocated to each link, depending on the number of nodes in the cluster. The larger the number of nodes the less the bandwidth allocated to each link; hence for each node. This value was chosen in accordance to the one used in [9]. A fixed simulation time was used so that the simulation mimics a more realistic scenario. It is assumed that between taking samples, the network is operating and packets are received and sent. Hence when taking a sample the bandwidth would have reduced and with a fixed simulation time, this varies only with the number of nodes in the cluster as follows:

Bused= 512(bytes/packet)\*4(packets/s)\*number of nodes in that cluster

Bavailable=70Mbps-Bused

Where 512bytes is the average packet size and the 4 packets/second is the transmission rate [10].

Table 1: Simulation Parameters

Parameter	Value
Area	10mX10m
Transmission range	20m
Number of Nodes	20 (increased to 50)
Transmission Bandwidth	70Mbps (for each Clusterhead)
Packet Size	512 bytes
Transmission rate	4 packets/second
Simulation Time	300s

### 7. Performance Analysis Of The Proposed Algorithm

A study was conducted on the proposed algorithm, to investigate its adaptability to node birth and node death. This investigation was done qualitatively, by examining how the Performance Factor (PF) varies with three parameters. The following analysis was carried out:

- Variation of Average Performance Factor (Avg PF) with the Number of Clusters
- Variation of Total Performance Factor (TPF) with the Number of Nodes Births
- Variation of Total Performance Factor (TPF) with the Number of Node Deaths (for Number of Node Deaths>MIN\_SIZE)
- Variation of Total Performance Factor (TPF) with the Number of Node Deaths (for Number of Node Deaths<=MIN\_SIZE)

#### 7.1 Variation of Average Performance Factor (Avg PF) with the Number of Clusters

Test data were generated for a fixed 20 node sources. A total of five samples were taken. Each time random nodes were generated and the number of clusters varied as a result. Depending on the number of clusters, the Performance Factor (PF) of each node, and hence each cluster also varied. The resulting graph is shown below:

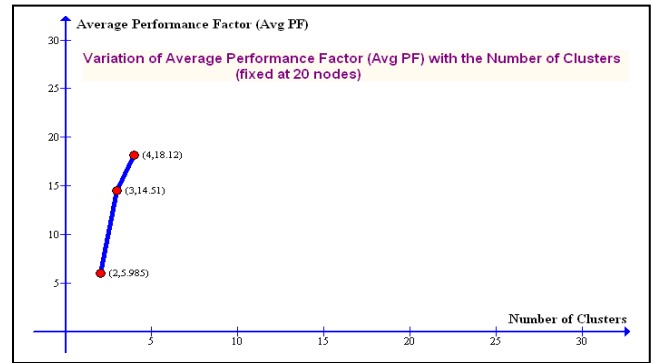


Figure 2. Graph of Avg PF vs. Number of Clusters

As can be seen from Figure 1 above, Average Performance Factor (Avg PF) increases steeply as the number clusters increases. This is because an increased number of clusters contain a lesser number of nodes for a fixed number of nodes and a greater amount of Average Bandwidth is available to each node. However, in all cases too many cluster results in inefficient use of resources as small clusters will result and a tradeoff must be made.

#### 7.2 Variation of Total Performance Factor (TPF) with the Number of Nodes Births

Another relation that was investigated is the variance of the Performance Factor with the number of node births in the network. Although, it is evident that as more and more nodes join the network, the Performance Factor decreases, however, the analysis shows that the decrease is not steep; for large number of nodes that join the network the aggregated PF does not change much. This is illustrated in the graph below.

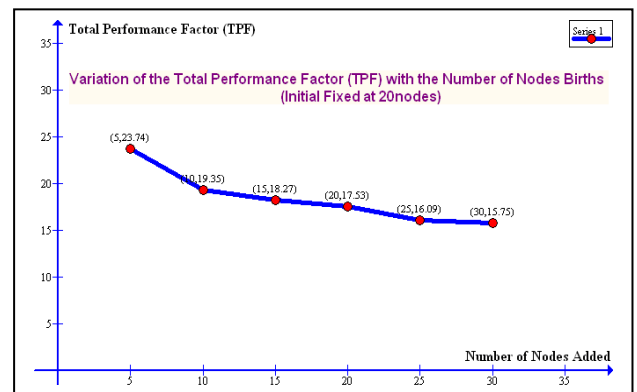


Figure 3. Graph of Total Performance Factor (TPF) vs Number of Node Births

The gradual decrease is of significant advantage provided by the algorithm as now more nodes can be accommodated into the network for a given PF. As before, the analysis is done on a network with an initial number of 20 nodes and incremented in steps of 5 nodes hereafter.

A critical evaluation of node deaths was carried out. Discussed before, a minimum cluster size is defined by using the variable MIN\_SIZE. As nodes die, the Cluster head determined if the number of neighbors it has is equal to or less than MIN\_SIZE. Depending on this value, two different

approaches are adopted. One section, investigates and justifies the use of MIN\_SIZE, in terms of Performance factor. Another one, investigates node births that results in cluster size to be greater than MIN\_SIZE, while the next section does the same for node births that is less than and equal to MIN\_SIZE.

**7.3 Variation of Total Performance Factor (TPF) with the Number of Node Deaths (for Number of Node Deaths>MIN\_SIZE)**

As can be seen from Table 2 above, as the number of node death increases while remaining above MIN\_SIZE, the PF rises significantly. This is because the number of nodes decreases while the numbers of clusters remain the same. It is deduced that such drastic increase in PF will result poor use of bandwidth, as now, too much bandwidth is now allocated to nodes that do not require such a high PF. It can be concluded that in such cases inefficient use of resources results.

**Table 2:** Variation of Total Performance Pactor (tpf) with the Number of Node Deaths (for Number of Node Deaths>Min\_Size)

Number of Node Deaths	Performance Factor (PF)	Total PF
5	4.53	22.787
	7.77	
	10.487	
10	6.1	24.5818
	8.2276	
	10.2542	
13	4.891	26.53
	8.249	
	13.39	
15	4.62	28.24
	10.02	
	13.6	

**7.4 Variation of Total Performance Factor (TPF) with the Number of Node Deaths (for Number of Node Deaths<=MIN\_SIZE)**

In light of the hypothesis made above, it becomes necessary to investigate the variation of PF as node deaths decreases the cluster size to a value less than MIN\_SIZE. This is done by reducing a specific cluster size to MIN\_SIZE or below for varying number of initial nodes in the cluster. Hence, the initial cluster size was varied from 20 to 50 nodes and dissolution of a specific cluster was carried out after which the PF of the remaining clusters were determined. As seen from Table 3 above, as nodes from too small clusters join the remaining clusters, the PF of the remaining clusters decreases. However, unlike the preceding section, inefficient use of bandwidth does not take place here. This is an important aspect of the algorithm as ad-hoc networks by

nature are extremely bandwidth constrained.

**Table 3.** Variation of Total Performance Factor (tpf) with the Number of Node Deaths (for Number of Node Deaths<=Min\_Size)

Test Nodes	Performance Factor (PF)	Total PF
20	3.2537	8.7101
	5.4564	
30	3.7983	12.0493
	8.251	
40	3.551	11.533
	7.982	
50	3.342	9.769
	6.427	

**8. Discussions and Conclusions**

All through the algorithm description till the performance analysis, the concept of adaptability of the scheme to node births and deaths have been addressed and tackled. The simulation and analysis shows that this adaptive scheme effectively adjusts to varying number of nodes in the network by using a performance metric that comprises of the bandwidth available to a node and its distance from the Cluster head. The graphs and the tables in Section 16 are a proof of how an optimum performance can be received by the nodes as the network scales in size. As a result, the algorithm also provides a tradeoff between the Performance Factor (PF) available to the nodes and the use of bandwidth, a crucial resource of any ad-hoc network.

**9. Future Work**

The concept of adaptability is a very important aspect of MANETs as well as other ad hoc networks and should be further explored. If adaptability can be efficiently tackled, it will provide with stronger grip on the randomness of MANETs, as a result of which the networks can be better managed. Even though, the proposed algorithm of this thesis effectively deals with the issue of adaptability, it can be enhanced to be more efficient. An important concern would be to design a scheme that decreases the performance factor after a small cluster joins its neighboring clusters, by an insignificant amount. Although merging to clusters prevents the misuse of bandwidth, a better trade off is needed so that PF of the existing cluster do not reduce too much; in such cases, the users might experience undesired performance level. Hence, this leaves room for tackling such problems, which should be focus of upcoming research.

**10. Acknowledgment**

We would love to thank first to Mr. Rashedul Hasan for providing me with the opportunity to work on this project which is a great source to gain invaluable experience in MANET as well as his valuable time and advice during our project. The experience that we gather is only possible due to his guidance and support.

## References

- [1] Abusalah, L. Khokhar, A. Guizani, M., "A survey of secure mobile Ad Hoc routing protocols," Univ. of Illinois at Chicago, Chicago, IL, USA, vol. 10, pp. 78-93, 2008.
- [2] S.Mangai, A.Tamilarasi, "Evaluation of the Performance Metrics in Improved Location Aided Cluster based Routing Protocol for GPS Enabled MANETs," European Journal of Scientific Research, Vol.46, pp.296-308, 2010.
- [3] Project on Wireless Adhoc Networking and Clustering, <http://www.cse.chalmers.se/~tsigas/Courses/DCDSeminar/Files/adhocproject.pdf>.
- [4] N. Chatterjee, A. Potluri and A. Negi, "A Self Organizing Approach to MANET Clustering," University of Hyderabad, International Conference on High Performance Computing, Dec. 2006.
- [5] P. Basu, N. Khan and T.D.C Little, "A Mobility Based Metric for Clustering in Mobile Ad hoc Network," Boston University, MCL Technical Report No. 01-15-2001.
- [6] R. Ghosh, A. Das, P. Som, R. Bhattacharya, P. Venkateswaran and S.K.Sanyal, "A Novel Optimized Clustering Scheme for Mobile Adhoc Networks," Jadavpur University, Poster in General Assembly of International Union of Radio Science (URSI), New Delhi, 2005.
- [7] Y. Wang and M, S. Kim, "Bandwidth-Adaptive Clustering for Mobile Adhoc Network," Washington State University, International Conference on Computer Communications and Networks, 2007, pp.103-108.
- [8] R.Misra, C.R.Mandal, "Performance Comparison of AODV/DSR On Demand Routing Protocols for Ad Hoc Networks in Constrained Situation", IEEE ICPWC, 2005.
- [9] S. R. Das, R. Castañeda and J. Yan, R. Sengupta, "Comparative Performance Evaluation of Routing Protocols for Mobile, Ad hoc Networks," LA , pp. 153- 161, October 1998.
- [10] P. Chenna Reddy, Dr. P. Chandrasekhar Reddy, "Performance Analysis of Ad-hoc Routing protocols," American Open Internet Journal, Vol. 17, 2006.

## Author Profile



**Neva Agarwala** received the B.S. in Electronics and Telecommunication Engineering from North South University, Bangladesh in 2008 and M.S. degree in Electrical and Electronic Engineering in 2009 from Imperial College London. Recently, Agarwala has pursued CCNA certification with outstanding score.

Currently, Agarwala is working as a Lecturer at Southeast University, Dept. of Electrical and Electronic Engineering.