CHAPTER 6

MESH CLUSTER APPROACH FOR CONNECTIVITY IMPROVEMENT

6.1 MANET CLUSTER

Mobile Ad Hoc Network represents complex distributed system with unpredictable node movements, result in frequent disconnectivity. Each node works independently, uses the resources based on their own desire. The main problem arises during the movement of the nodes and random utilization of network resources. This work attempts to solve the mobility maintenance issues with three mesh structures; i) Mesh Tree ($M_T$), ii) Mesh Backbone ($M_B$) and iii) Mesh Cluster ($M_C$). This work shows the various mobility maintenance architectures with connectivity improvement analysis. The performance of the proposed work is analyzed through mathematical models and simulation results. The relation between connectivity and computing parameters are also addressed in this work.

6.1.1 Mesh Cluster Approaches

The mesh cluster network architecture is based on; 1. Greedy algorithms and NP hard problems related to connectivity, 2. Virtual Ad Hoc structures and 3. MANET and Mesh structures. Hence the related work deals with the above mentioned areas. Greedy algorithms (Li et al. 2005a & 2005b) used to find the minimum weights in stable structures when the link weights or cost values are taken from finite range. But for the optimum result in mobile scenario, greedy requires localized structures for alternate routing and connectivity loss. This type of constructions is used in our proposed work.
Recently Fedor et al (2009) proved that, any graph $G$ and integer $k \geq 0$, either finds a spanning tree with at least $k$ internal vertices, or outputs a new graph $G_R$ on at most $3k$ vertices. Such that an integer $k^1$ in $G$ has a spanning tree with at least $k$ internal vertices if and only if $G_R$ has a spanning tree with at least $k^1$ internal vertices.

Clustering (Yang et al 2005) is a process that divides the network into interconnected substructures, called clusters. Each cluster has a Cluster Head (CH) (Marco & Garcia 2007) as coordinator within the substructure. Each CH acts as a temporary base station within its zone or cluster and communicates with other CHs. In a clustering scheme, the MANET is firstly partitioned into a number of clusters. A cluster head is then allocated for each cluster which will perform various tasks on behalf of the members of the cluster. Better connectivity among the cluster-heads is an advantage for applications such as message broadcasting. The use of independent dominating sets (Rajesh & David 2006) as cluster-heads is problematic when the network topology changes. In particular, two cluster-heads move within the transmission range of each other is given by (Shen et al 2004), one of them must defer to the other which can trigger cluster-head changes that may propagate throughout the network.

6.1.2 Cluster Algorithms and Complexity

The solution for NP hard Problems are obtained based on a partitioning of the network or problem domain. During the search, it eliminates some of the branches along with complexity of $O(2^n)$ for constant $0 < c < 1$, where $n$ is the output size of the problem. Every mobility based NP-complete problems can be solved by exhaustive search or by introducing faster algorithms. Unfortunately, when the size of the instances grows the running time for exhaustive search soon becomes large, even for small
instances. Tian & Georganas (2005) prove that, the communication range is twice of the sensing range in wireless sensor network. This is motivated us to develop a coverage area based localized structure for mobility maintenance in ad hoc domain.

The number of neighbors required to maintaining the connectivity and capacity of the network is studied from Gupta & Kumar (2000). In a network with n randomly placed nodes, each node should be connected to O (log n) nearest neighbors. If each node is connected to less than 0.074 log n nearest neighbors then the network is asymptotically disconnected, while if each node is connected to more than 5.1774 log n nearest neighbors then the network is asymptotically connected with probability approaching one as n increases. The access point based connectivity is studied from Shang (2009), in which L, r > 0 and variables V₁,…,Vₙ are uniformly distributed in the interval of [0, L]. The vertex set with edge VᵢVⱼ, if |Vᵢ − Vⱼ| ≤ r and some access points {zⱼ} can exist in G (n, L). This concept is well suitable for infrastructure based wireless networks. Similarly connectivity of one dimensional graphs also studied from (Boegsterd et al 2011, in which the graph denoted by Gₓ (n, L) if there exists a fixed node at the point x ∈ [0,L]. In particular, when x =0, the graph is G₀(n, L).

6.1.3 Coverage Area Approaches

Cover set based algorithms divide the nodes into cover sets, where each cover set is capable of monitoring all targets. Generating the maximum number of cover sets is an NP-complete problem and, thus, algorithms producing sub-optimal solutions have been proposed. Dimitrios et al (2010) proposed a coverage algorithm that can produce both disjoint coversets, i.e. coversets with no common nodes, as well as non-disjoint coversets. To include in a coverset, algorithm uses a cost function that takes into account
the monitoring capabilities of a node, its association with poorly monitored targets, but also the nodes remaining battery life. This approach is motivating us to work with coverage based concept in mesh environment.

6.1.4 Energy Efficient Heterogeneous Clustered Scheme

Dilipkumar et al (2009), Bao & Garcia (2010) analyze the heterogeneity of nodes in terms of their energy and hierarchical cluster. Homogeneous clustering protocols assume that all the sensor nodes are equipped with the same amount of energy and as a result, they cannot take the advantage of the presence of node heterogeneity. Adapting this approach, an energy efficient heterogeneous clustered scheme is introduced for wireless sensor networks based on weighted election probabilities of each node to become a cluster head according to the residual energy (Chang et al 2000) in each node. There are three common types of resource heterogeneity in sensor node: computational heterogeneity, link heterogeneity and energy heterogeneity. Computational heterogeneity: Heterogeneous node has a more powerful microprocessor and more memory than the normal node. Link heterogeneity: The heterogeneous node has high-bandwidth and long-distance network transceiver than the normal node. Link heterogeneity can provide more reliable data transmission.

6.1.5 Cluster-Based Topology Control

Xu et al (2008) proposed a research tool for static and dynamic topology control in network simulator 2. Static topology control algorithm is based on network scenarios. Transmitting power is assigned to each node before the operation of the network. Therefore, the simulation of static topology control in Network Simulator is correspondingly simple as the algorithm can be run independent of the simulation environment. Dynamic topology control is relatively complex due to node failure or mobility which
results in necessary changes of topology to keep an appropriate transmitting power for each wireless node throughout the lifetime of the network.

Tao et al (2007) proposed a hybrid approach for topology control. It uses a centralized algorithm within a cluster and to achieve strong connectivity, localized information exchange between adjacent clusters. Specific instances of the topology control problem are studied by considering two optimization functions: Minimizing the maximum power used by any node in the network. This criterion is referred as MINMAX and maximizing the total power used by all of the nodes in the network. This is equivalent to minimizing the average power used by the nodes. This criterion is referred to as MINTOTAL.

The CLTC framework consists of three phases: Nodes autonomously form clusters and elect cluster heads. Using the information gathered from the members of its cluster, each cluster head assigns transmission powers to the nodes in its cluster such that strong connectivity is guaranteed within each cluster. Each cluster head selects a set of border nodes, shares information with neighbouring cluster heads and computes appropriate transmission powers for border nodes in order to establish strong connectivity with the neighbouring clusters.

6.2 NETWORK FORMATION

The network construction is as follows:

A simple heuristic based on coverage area is used to create the mesh structure in ad hoc network. A node is able to cover ‘K’ neighbors within its coverage area it follows mesh topology.

I. Mesh Tree Formation
\[ M_T = \sum T_i(v_i) + M_i(v_j) \mid i,j=1,2,3,\ldots \] (6.1)

where \( v_i \) – root vertex, \( T_i \) – leaf of \( v_i \) or Tree of \( v_i \), \( M_i \) – mesh structure of intermediate nodes \( v_j \).

Subject to:

\[ T_i \neq M_i \quad (6.2) \]

\[ T_i < A_i < M_i, \quad i \neq 0 \quad (6.3) \]

\( A_i \) – alternating request.

II. Mesh Backbone Formation

The following relation between \((b, l, e)\) is used for network construction in graph \(G\), where \(b\)-Mesh backbone (\(B\)-backbone set, \(B \subseteq G\)), \(l\) – leaf node, \(e\)-edge between nodes. The objective of the construction is with minimum value of backbones network connectivity is maintained. Assume ‘\(n\)’ number of backbone node ‘\(b\)’ is required to include with the network size of \(N\), where \(n \gg N\).

\[ \min B \Rightarrow \sum b_e l_e \quad \text{subject to:} \]

\[ \sum b_e \leq G(v), \quad e \in E, v \in V \quad (6.5) \]

\[ \sum b_e = n B, \quad e \in E \quad (6.6) \]

where ‘\(n\)’- number of nodes per backbone structure \(n < b_e \)

\[ \sum l_e + \frac{1}{2} e = B_1 \text{ (New backbone)} \quad (6.7) \]

III. Mesh Cluster Formation
Clusters are formed by dividing the entire area into small regions. Clusters heads are formed based on the selection of edge nodes of each region and able to form the mesh structure. Each node in the cluster is hierarchically arranged.

\[
\text{Graph } G = n^* M_c \quad (6.8)
\]

\[
M_c (n) = \sum C_n + \sum L_n \mid n=0,1,2,3,\ldots, 
\]

Subject to:

\[
M_c - \text{mesh clusters, } n - \text{number of nodes in each cluster,}
\]

\[
C_n, L_n - \text{cluster and leaf nodes. Subject to:}
\]

\[
0 < M_c / C_n + L_n < 1 \quad (6.10)
\]

\[
C_n \neq L_n \quad (6.11)
\]

\[
C_n \leq \sum L_n \quad (6.12)
\]

6.3 MESH TREE CONSTRUCTION

A mesh tree is a combination of localized tree with mesh structure. A simple approach is used to construct the mesh structure from the unconnected network. Initially randomly a root node is selected and named as \( V_R \) (Root Vertex). The edge of \( V_R \) is \( e_i \) \( [(i=1,2,3\ldots,n)] \), this value due to multiple \( e_i \) of \( V_R \), is added to \( v_i \) (one hop neighbor in any direction). After this process \( e_i \) of \( v_i \) is added to \( v_j \) until the entire nodes are connected to \( V_R \) through \( v_i \) and \( v_j \) \((i,j=1,2,3\ldots n)\). During this process to avoid looping problem, edge of \( V_R \) uses unique ID. The nodes process only the new path ID and do not process the existing ID. This process is used to prevent looping problem. This feature is also supported by ad hoc on demand routing protocols. After
the creation of tree structure based on the coverage area of \( v \), mesh clients are identified. A node is able to connect more than three nodes which are named as mesh client.

The entire network construction and link failure scenario is shown in Figure 6.1. Once a network is constructed, maintenance and alternate routing is an interesting work, especially with mobile clients. To maintain the connectivity mesh topology nodes are identified in the desired forward and reverse directions. From the figure, mesh structures are formed based on the coverage area; named as \( M_{s1s2} \). If the link or node failure occurs forward direction \( M_{s1} \) or reverse direction \( M_{s2} \) is used for alternate routing. Normally forward direction structure is the first option for node link failure and reverse direction is the second option if first link fails. In the way all the nodes connectivity is maintained in the network.

![Mesh tree constructions](image)

**Figure 6.1 Mesh tree constructions**

### 6.3.1 Mesh Tree Algorithm
Input: Unconnected graph G with vertices V and edge E // V ← v₁, v₂, ..., vₙ

Output: Tree with mesh structure (nodes with mesh topology)

/* Mesh topology construction */

1  Select root vertex Vₘ in G i.e. Vₘ ∈ G

2  Add Edge eᵢ of Vₘ to vᵢ, and vᵢ to vᵢ until all the nodes connected to Vₘ through intermediate nodes, where R, i = 1, 2, ..., n

   // the variables 1, 2,...n indicate if one node fails or change the position another node act a root vertex based on high coverage area

   // tree formation process, to avoid looping problem each ei is assigned with unique ID.

3  if edge of vᵢ form a loop with Vₘ or between the vertices then

4  Construct a localized mesh Mₛ, i.e. more than three vᵢ’s are in the same coverage area // v₁, v₂ and v₃ ∈ Mₛ, v₁, v₂ and v₃ form the mesh topology

5  else

6  go to step 2

/* Structure Maintenance and alternate routing */

7  if vertices v₁ not connected to V (any node in V) and Vₘ or link lᵢ is disconnected then

8  Alternate route is obtained via Mₛ₁ // where Mₛ₁ be first level mesh structure in the forward direction toward the desired node

9  if Mₛ₁ fails
10  \( v_1 \) try \( M_{s2} \) // where \( M_{s2} \) be first level mesh structure in the reverse direction toward the desired node

11  else

12  wait for connectivity or go to step 2

13  end

6.3.2 Correctness of the algorithm

To prove the correctness of proposed algorithm, the following arguments are analyzed with proof: i) in line 2: Construct tree and loop (mesh structure) and ii) in line 7 and 8, if link fails connections are maintained through \( M_{s1} \) and 2. iii) in line 10, if \( M_{s1} \) fails, \( v_1 \) try \( M_{s2} \).

For i), Tree structure formation based on vertices and their edges added with one hop neighbors, it has been proved as follows; if a graph \( G \) is connected, tree is build first, then find an mesh structure based on the selection of nodes are in the same coverage area, and connect all nodes, a mesh tree is formed. Mesh tree is a combination of loops and tree, where loops are used for alternate routing if tree fails.

For ii) Vertices \( v_1 \) is not connected to \( V \) (any node in \( V \)) and \( V_R \), connectivity is maintained through \( M_{s1} \& M_{s2} \). Connection of any \( (v_1) \) is maintained via \( M_s \) means that there is always a possible node degree between \( v_1 \) and \( M_s \) within the Mesh structure. This is true because \( G \) is a Mesh Tree (T), with sufficient node degree. If one node degree fails or removed from graph (G), network structure is maintained with one hop mesh topology node.

For iii) if \( M_{s1} \) fails, \( v_1 \) try \( M_{s2} \). \( G \) be a combination of loops and tree, subset \( V \) (\( v_1 \)) + \( V' \) (\( M_s \)) = \( G \), such that \( G-V'=V \). And also \( G \) be a tree, a
connected graph with deleting any edge cannot be disconnected to it. Hence it is possible to add $v_1$ to $M_{s2}$ or $M_{s1}$. It proves final correctness.

6.4 MESH BACKBONE FORMATION ($M_B$)

The second approach is mesh backbone, unlike the previous approach there is no tree structures for mobility maintenance. In the previous work tree maintenance process requires a mesh structure, where in this approach there are no restrictions for mesh backbone construction and it is constructed as follows; when a node has more than two (fully) connected neighbors it is marked as mesh backbone node. If mesh backbone connectivity fails, connections are maintained through neighbor leaf node. The construction process is given in Figure 6.2. In the figure red colored node is able to connect mesh structures with clients 3 and 4. During the movement of the backbone node, based on the leaf nodes coverage area connectivity maintained and leaf node becomes backbone node.

![Figure 6.2 Mesh backbone constructions](image-url)
6.4.1 Mesh Backbone Construction

Assume ‘n’ number of backbone node ‘b’ is required to include with the network size of N, where n>>N. we use the following relation (b, l, e) for network construction in graph G, where b-Mesh backbone (B-backbone set, B∈G), l – leaf node, e-edge between nodes. The objective of the construction is with minimum value of backbones network connectivity is maintained. The minimum size of the backbone, i. e. min B = Σ b_i l_e, subject to b_e and l_e values less than the total number of vertices in G. The network construction algorithm is as follows;

(i) Select the mesh backbone node in G based on the definition given in section 4
(ii) Let b_i is the backbone node with more than two connected neighbors l_i
(iii) Connect b_i and l_i, i=1,2,...n // mesh backbone construction
(iv) Remove b_i with l_i // disconnect backbone with leaf node
(v) Maintain l_i via l_j // l_j be the another leaf node attached with b_j
(vi) l_j become a new backbone.

6.4.2 Theoretical Analysis

To prove the effectiveness of the proposed work the following theorem is mathematically analyzed.

Theorem 4: A mesh backbone node able to connect or delete at least n nodes in one hop.
Proof: Let \(v_1, v_2, v_3, \ldots, v_k\) be the mesh backbone nodes in the network, the total non backbone nodes in the network is given by \(N_v^* \times H\) (\(N_v\)- non backbone nodes, \(H\)- hop count) for selection of \(n\)-hop backbone connections. Consider one hop connection between backbone nodes. The maximum number of backbone node connectivity of the \(i^{th}\) system is given by \(V_i \geq N_v \times (i+1)\) for \(i=1, 2, 3\) (\(\Delta\)- node degree). Connectivity of a network is defined as the probability of the number of one hop backbone connecting from one non mesh backbone to another node in the graph. Let \(s\) be the sleeping node, then \(s_{n1}, s_{n2}, \ldots, s_{nn} \in G(V, E)\) otherwise \(s_{n1}, s_{n2}, \ldots, s_{nn} \cup G(V, E)\). Hence the virtual node connects or deletes links in one hop.

**Lemma 1:** The mesh backbone structure computes active node, sleeping node set of \(G\).

Proof: From theorem 4, mesh backbone contains a dominating set, active and sleeping node set of \(G\). Let \(u\) and \(v\) be a pair of vertices of \(T\). The path between \(u\) and \(v\) are \(q\) distance and both are connected with \(b\) (next level virtual node). This is the shortest one hop path between \(b\) and other nodes. In this way all \(b\) nodes say \(b_1, b_2, b_3, \ldots\ etc\), are connected to corresponding nodes \(a\) and \(s\), where \(a, s\) are the active and sleep nodes. This also supports the connectivity and deletes connection from \(v\) to \(u\). Thus the mesh backbone used to compute active and sleep nodes of \(G\). Hence the lemma follows the theorem 4.

6.5 MESH CLUSTER APPROACH (MC)

Cluster based routing is one of the solution for reducing the control load in large scale networks. In this work it is used to create a mesh structure for improve the connectivity performance.
Figure 6.3 Cluster mesh structure construction

The network construction process is as follows: initially divide the entire area into sub clusters. Select the cluster head which is closer to the nodes in nearby region (this process used to connect other nodes in the neighboring cluster). Mesh structure is formed between cluster heads and inside the cluster region. Similarly select the secondary cluster head for alternate routing. The mesh construction process is shown in Figure 6.3. From the figure different coloring scheme is used to differentiate cluster heads and other nodes. The colors red, purple and blue are allotted for clusters, secondary clusters and leaf nodes respectively. The entire routing request is performed through the cluster heads. This structure is used to reduce the control load and improve the connectivity even the nodes are in mobile.
6.5.1 Algorithm for Mesh Cluster Construction

1: Input: Network C, with N nodes // N is the network size
2: Output: Mesh cluster \( m_c \) with \( n \) nodes, where \( m_c \in C \) and \( n \in N \) // \( n \) is the cluster size
3: Initialization: Selection of cluster head \( H_{vc} \)
4: if C is divided into \( c \) clusters of different size, \( c \in C \), \( c \) is non-empty, node x near to node y // node x and y are from different cluster, but nearby nodes.
5: x send \( H_{vc} \) hello packet in c
6: Wait for t seconds // listen the medium
7: No response received, \( x=H_{vc} \) // \( x \) and \( y \) be the cluster head
8: else
9: if response received from node z
10: \( x \) knows \( z=H_{vc} \) // already existing cluster head in c
11: \( x \) join with \( z \)
12: end

Cluster structure formation is based on hierarchical architecture as follows: 1. Cluster head with high node degree, 2. secondary cluster head nodes and 3. Leaf node (attached with any one of the above). The hierarchical order provides many connections in one hop and performs the alternate routing during movement. After the cluster structure construction, each node involves in routing and information processing. The sleep node information is maintained by the cluster head and it is used for topology control and increase the network life time. To maintain the topology and connectivity, leaf nodes are resumed by the cluster head when it is required. This process
mainly supports the low power consumption of each node and also used to improve the network life time. Finally normal node (also named as leaf node) involves only information exchange process and it will become mesh cluster node when the existing cluster head fails. In the proposed work, node degree is counted only from the cluster node to leaf nodes, not between the cluster nodes. Most of the existing research work restricts the leaf nodes movement i.e. it must roam inside the cluster head's coverage area. But in this work, there is no restriction for leaf nodes, once it comes out of the cluster it join with other clusters.

The computation complexity of cluster algorithm is \(O(n^2 \log n)\), resulting from network partition and cluster head election. In the cluster algorithm one head node may potentially need to be executed as many as \(O(n)\) times for each of the \(O(n)\) leaf come out from \(v_c\). Hence the complexity initially in the order of \(O(n^2)\) and \(\log n\) is added to this complexity if the nodes are from other clusters. For finding the lowest complexity of the minimal \(v_c\), \(O(n + \Delta \log n)\) is required, where \(n\) is the number of nodes in graph \(G\) and \(\Delta\) - node degree.

6.6 RESULT AND DISCUSSION

In this simulation each host is modeled as an infinite-buffer, store-and forward queuing station. The wireless LAN (Local Area Network) Distributed Coordination Function (DCF) with Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) is used as the medium access control protocol. All the simulations are conducted using ns-2.30 with 200 nodes with a 1000x1000 m\(^2\) deployment area with random way point model is used. The number of nodes ranges from 20 to 200 with increment step of 20, which allows us to evaluate these protocols on increasingly dense networks with average node degree ranging from 2 to 10 for all the algorithms.
The simulation results are average values taken from 10 communication pairs. In this approach, broadcast transmission is allowed between mesh clusters. One hop reserved cluster head and leaf/normal nodes are permitted to send only unicast transmission messages. These choices can reduce the possible packet collisions and save node energy. Node degree is a measure between forward and downlink to nearby nodes. Hop count are intra mesh cluster based i.e. inside the mesh cluster hop count is named as $H_{ms}$ (Hop stretch) and normal hop of the graph is original $H_s$. Parameters are analyzed based on mesh nodes and clusters.

6.6.1 Performance Metrics

In the proposed approach virtual structure concentrates on the issues of cluster backbone with stable and mobile routing. The following parameters are used to analyze its performance.

- **Protocol duration**: It is measured as the time taken by the method to finish backbone formation.

- **Hop stretch ($H_s$)**: Ratio of number of hop count by mesh cluster to the original hop count between any two nodes. Hop stretch between the nodes $u$ and $v = H_{ms}(u, v) / \text{original } H_s$ of $(u, v)$.

- **Transmission range vs. CDS**: Transmission range varies from 100m to 300m and different structures performance are plotted.

- **Power consumption**: Average power consumption per node is analyzed in this case.
• Node degree vs. Network size: Initially network was formed with different node degree and the structure is named as virtual clusters; node degree will be increased in the order of seven per node and a total of 30 will be added to local cluster structure and the percentage of mesh cluster requirements in terms of node degree is plotted in the result analysis.

• Control load: The control load analysis is based on number of broadcasting packets required to connect the nodes with the cluster.

6.6.2 Simulation Results

Cluster head robustness is defined as the number of cluster member nodes whose disappearance will make the cluster head disconnected. The robustness of cluster head is shown in Figure 6.4. The robustness deals with the failure of cluster head with the connectivity performance of the nodes. From the robustness graph mesh cluster outperforms other two approaches for large scale network size of 200 nodes. But up to network size 40, all the three approaches show the same performance. The mesh cluster approach withstand even the node fails more than ten i.e. if 14 nodes are not connected with the structure, cluster head will fails. Due to more number of nodes attached in the structure, mesh cluster head maintain its connectivity with other nodes for large network size. However, the robustness of the Mesh Tree is suffered with leaf nodes mobility. In the same manner, Mesh Backbone connectivity is based on backbone header. From the analysis if more than 10% of node fails, Mesh backbone shows poor connectivity.
Similarly other methods cannot support the connectivity below 5%. Due to minimum unicast packet flow per node, average power consumption per node of Mesh Tree approach is low compared with existing methods and the result is shown in Figure 6.5. From the comparison graph mesh backbone requires optimum power consumption i.e. half the value required by mesh cluster approach. This is the default value 0.3 W used in network simulator-2. Mesh cluster requires more power due to the information processed by most of the member nodes. The protocol duration for these three protocols is measured as the time used by them to finish backbone formation. Fusion Backbone requires initially 2 seconds for 30 nodes and reaches the maximum value of 3.8 seconds for 200 nodes. The zone and CBA approaches formed by less than 2.5 sec for 200 nodes. The protocol duration is gradually raised when the network size is increased. If local computations is minimum,
protocol duration is maintained constantly. From the Figure 6.6, mesh tree takes less time than other approaches.

![Figure 6.5: Power consumption for various network size](image)

For small scale networks, mesh tree and mesh backbone even requires less time than mesh cluster. But tree approach outperforms in dense network scenario, again due to its high degree of localization. Note that, for mesh cluster and backbone approaches, most of the time spent for electing the headers and backbones, through sub graph formation, which shows a lower degree of localization. Mesh tree requires one second for tree head formation where the backbone and cluster based approaches need 2.5 sec and 3.8 sec respectively. Hop stretch between the nodes \( u \) and \( v \) = CDS \( H_{ms} (u, v) \) / original \( H_s \) of \( (u, v) \). Hop stretch for a graph \( G \) is calculated from greedy algorithm. The \( H_s \) for the three approaches is given in Figure 6.7. For all the methods \( H_s \) lies between 1-1.6, mesh backbone requires minimum hop count due to minimum degree backbones and others require localized/distributed computation with maximum hop count.
Figure 6.6 Protocol duration for cluster based approach

However as mentioned above, the message size of tree protocol is unbounded. Since every node needs to exchange its neighbor list with its own neighbor, this message size will become very large in dense networks.

Figure 6.7 Hop count ratio between different mesh structures
Moreover, in practice, the loss rate of large packets will be much higher than that of small packets. Note that, the sizes of all the messages used by mesh cluster and backbone algorithm are bounded by some constants. Comparing these protocols gives the difference in message overhead. The proposed protocols are more significant than that of other approaches. The reason is that other systems heavily use unicast packets which cause lots of acknowledgements and retransmissions at MAC layer. The control load requirements for the proposed methods are shown in Figure 6.8. From the graph, mesh tree requires more message overheads due to leaf maintenance and more broadcasting packet flow between the tree header and other nodes. The other approaches requires minimum control load, since its follows distributed localized header and backbone formation.

Figure 6.8 Message overhead for backbone protocols

Among these algorithms, CBA performs best in term of less number of mesh tree because it constructs the head nodes with maximum
degree. The node degree for three methods is given in Figure 6.9. Node degree requirement for the mesh cluster is vary from 2 to 3. The other method backbone needs double the size of node degree for 200 nodes, since it follows flat network architecture, where cluster is based on hierarchical structures with no restriction on cluster formation. It is purely based on the coverage area of the cluster head and the leaving nodes act as backbones for short time to maintain the connectivity.

![Figure 6.9 Node degree requirements for different network size](image)

6.7 CONCLUSION

The mesh cluster is a method by which the hosts are hierarchically organized on the basis of the proximity. The hierarchical structure forms the personal communication networks so that the hosts can be locally organized. This work, first proposed a tree based protocol for solving the looping problem and secondly an approximation algorithm based on backbone and cluster for wireless Ad Hoc networks. The proposed clustering algorithm is a
distributed implementation of virtual structure with inter and intra cluster domains. This chapter shows an upper bound on the time and message complexity for finding a near optimal cluster-head set size. And also node level computations is a very challenging issue in fast moving distributed dynamic networks to solve connectivity problems. This work defines a model for mobile Ad Hoc network and supports connectivity maintenance among nodes.